

1. 如何阅读本章

该I²C总线模块的介绍适用于所有LPC111x系列芯片。

2. 特性

- 标准的I²C兼容总线接口可以配置为主模式、从模式或主/从模式。
- 当总线上无串行数据时，多个主设备同时传输则会进行仲裁。
- 允许对时钟编程，以调整I²C的传输速率。
- 数据在主设备和从设备之间的传输是双向的。
- 串行时钟同步使得不同位速率的设备可通过一个串行总线进行通信。
- 串行时钟同步作为握手机制来挂起和恢复串行传输。
- 支持增强型快速模式（Fast-mode Plus, FMP）。
- 从设备可多达四个不同的地址。
- 监测模式允许观察所有I²C总线的通讯，忽视从机地址。
- I²C总线可用于测试和诊断。
- 在I²C总线包含一个带两个引脚的标准的I²C兼容总线接口。

3. 应用

该接口用于与外部I²C标准器件连接，如串行存储器、LCD、音调发生器和其他微控制器等。

4. 基本描述

典型的I²C总线配置如图 10-16所示。根据方向位的状态（读/写），在I²C总线上可能有两种类型的数据传输：

- 数据传输从一个主设备的发送器到一个从设备的接收器。主设备发送的第一个字节是从设备的地址。紧接着的是大量的数据字节。从设备在每个字节接收完成后返回一个应答位。
- 数据传输从一个从设备的发送器到一个主设备的接收器。由主设备发送的第一个字节（从设备的地址），然后返回一个应答位，紧接着的数据字节由从设备传输到主设备。除了最后一个字节之外，主设备在其他每个字节接收完成之后返回一个应答位。在最后一个字节接收完成后，返回一个“非应答”。主设备产生的所有的串行时钟脉冲与

起始（START）和停止（STOP）条件。一次传输由STOP条件或重复START条件来结束。由于重复START条件将开始下一个串行传输，故I2C总线不会被释放。

I2C接口是面向字节的，有四个工作模式：主发送模式、主接收模式、从发送器模式和从接收模式。

I2C接口符合完全I2C规范，可以将ARM Cortex - M0处理器断电而不干扰同一个I2C总线上的其他设备。

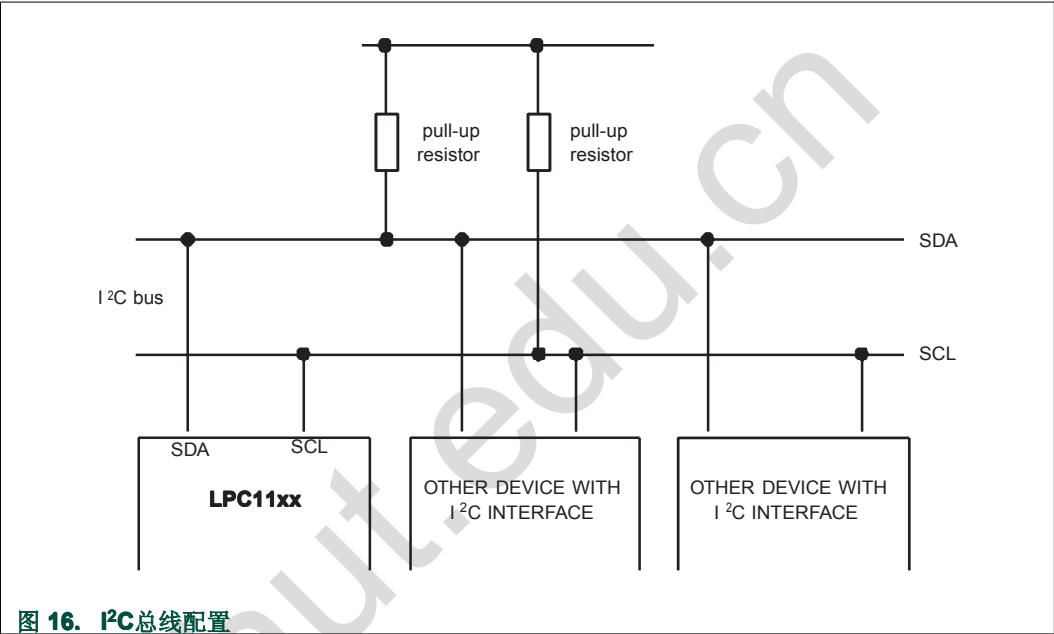


图 16. I2C总线配置

4.1 I2C 增强型快速模式（Fast-mode Plus）

增强型快速模式支持1兆位/秒的传输速率，用于与恩智浦半导体现在提供的I2C总线设备进行通信。

为了使用增强型快速模式，必须在IO CONFIG寄存器块中正确地配置I2C引脚，见 [表 7-73](#) 和 [表 7-74](#)。在增强型快速模式中，速率可大于400 kHz和最高可达1 MHz。

5. 引脚描述

表 137. I2C总线引脚描述

引脚	类型	描述
SDA	输入/输出	I2C 串行数据
SCL	输入/输出	I2C串行时钟

I2C总线引脚必须通过IOCON_PIO0_4寄存器 ([表 7-73](#)) 和IOCON_PIO0_5寄存器([表 7-74](#)) 将其配置为标准/快速模式或增强型快速模式。在这些模式中， I2C总线引脚是开漏输出，完全兼容 I2C总线规范。

6. 时钟和功耗控制

I2C总线接口的时钟（PCLK_I2C）是由系统时钟（见 图 3-3）。这个时钟可以通过 AHBCLKCTRL 寄存器 (3-4.14节) 以节省功耗。

7. 寄存器描述

表 138. 寄存器概览: I²C (基地址 0x4000 0000)

寄存器名	访问方式	偏移地址	描述	复位值 ^[1]
I2C0CONSET	R/W	0x000	I²C 控制设置寄存器. 当写入1到这个寄存器的某位，则I ² C控制寄存器相应的位被设置。写0对I ² C控制寄存器相应的位没有影响。	0x00
I2C0STAT	RO	0x004	I²C 状态寄存器. 在I ² C工作过程中，这寄存器提供了详细的状态码，以帮助软件确定下一步的动作。	0xF8
I2C0DAT	R/W	0x008	I²C 数据寄存器. 在主或从发送模式下，数据在写入这个寄存器后被传送。在主或从接收模式，已收到的数据可从该寄存器读取	0x00
I2C0ADR0	R/W	0x00C	I²C 从地址寄存器 0. 当 I ² C 接口处于从模式时，该寄存器包含 7 位的从地址；在主模式时无用。最后一个有效位决定从设备是否响应广播地址。	0x00
I2C0SCLH	R/W	0x010	SCH 占空比寄存器高半字. 决定 I ² C 时钟高电平时间。	0x04
I2C0SCLL	R/W	0x014	SCL 占空比寄存器低半字. 决定 I ² C 时钟低电平时间。I2nSCLL 和 I2nSCLH 共同决定由 I ² C 主设备产生和从设备某些时候使用的时钟频率。	0x04
I2C0CONCLR	WO	0x018	I²C 控制清零寄存器. 当写入1到这个寄存器的某位，相应的I ² C控制寄存器的位被清零。写一个0对相应的I ² C控制寄存器位没有影响。	NA
I2C0MMCTRL	R/W	0x01C	监测模式控制寄存器.	0x00
I2C0ADR1	R/W	0x020	I²C 从地址寄存器 1. 当 I ² C 接口处于从模式时，该寄存器包含 7 位的从地址；在主模式时无用。最后一个有效位决定从设备是否响应广播地址。	0x00
I2C0ADR2	R/W	0x024	I²C 从地址寄存器 2. 当 I ² C 接口处于从模式时，该寄存器包含 7 位的从地址；在主模式时无用。最后一个有效位决定从设备是否响应广播地址。	0x00
I2C0ADR3	R/W	0x028	I²C 从地址寄存器 3. 当 I ² C 接口处于从模式时，该寄存器包含 7 位的从地址；在主模式时无用。最后一个有效位决定从设备是否响应广播地址。	0x00
I2C0DATA_BUFFER	RO	0x02C	数据缓冲寄存器. 在每次从总线上接收9位数据（8 位数据加 ACK 或 NACK）后，自动将移位寄存器I2DAT的前八位的内容传送到 DATA_BUFFER 寄存器。	0x00

表 138. 寄存器概览: I²C (基地址 0x4000 0000) ...续

寄存器名	访问方式	偏移地址	描述	复位值
I2C0MASK0	R/W	0x030	I²C 从地址屏蔽寄存器 0. 这个屏蔽寄存器与 I2ADR0共同确定一个地址匹配。该寄存器对于地址与广播地址 ('0000000')比较时没有影响。	0x00
I2C0MASK1	R/W	0x034	I²C 从地址屏蔽寄存器 1. 这个屏蔽寄存器与 I2ADR0共同确定一个地址匹配。该寄存器对于地址与广播地址 ('0000000')比较时没有影响。	0x00
I2C0MASK2	R/W	0x038	I²C 从地址屏蔽寄存器 2. 这个屏蔽寄存器与 I2ADR0共同确定一个地址匹配。该寄存器对于地址与广播地址 ('0000000')比较时没有影响。	0x00
I2C0MASK3	R/W	0x03C	I²C 从地址屏蔽寄存器 3. 这个屏蔽寄存器与 I2ADR0共同确定一个地址匹配。该寄存器对于地址与广播地址 ('0000000')比较时没有影响。	0x00

[1] 复位值只反映被使用位的数据存储情况，不包括保留位的内容。

7.1 I²C 控制设置寄存器 (I2C0CONSET - 0x4000 0000)

I2CONSET寄存器控制I2CON 寄存器（用于控制 I²C 接口运作）的位设置。写入1到该寄存器的某位，将导致I2C控制寄存器的相应位被设置。写0则没有影响。

表 139. I²C 控制设置寄存器 (I2C0CONSET - address 0x4000 0000) 位域描述

位	符号	描述	复位值
1:0	-	保留，用户软件不要对保留位写‘1’。从保留位读取的值没有定义。	NA
2	AA	生效应答标志位。	
3	SI	I ² C 中断标志位。	0
4	STO	STOP 标志位。	0
5	STA	START 标志位。	0
6	I2EN	I ² C 接口允许。	0
31:7	-	保留，从保留位读取的值没有定义。	-

I2EN I²C 接口允许。当 I2EN为1时，I²C接口被允许。写“1”到 I2CONCLR 寄存器的 I2ENC 位能清除I2EN位。当 I2EN 为 “0”时，I²C接口被禁止。

当I2EN为 “0”时，SDA 和 SCL的输入信号被忽略，I²C 处在 “不可寻址地址” 的从设备状态，且STO位被强制为 “0”。

I2EN不应被用来暂时释放I²C总线，因为当I2EN 被复位时I²C总线的状态会丢失，而应用 AA 标志位来代替

STA 是 START 标志位。设置这个位导致I²C接口进入主设备模式，并传输一个START 条件，如果I²C接口已经是主模式则传送一个重复 START条件。

当 STA 为 1 而且I²C接口还不是主模式，它将进入从模式。然后检查总线，如果总线空闲则产生一个START条件；如果总线忙，它将等待STOP条件(它将使总线空闲)并在内部时钟发生器延时半个时钟周期后产生一个 START 条件。如果I²C已经是主模式而且数据已经

发出或者接收到，它将发送一个重复START 条件。可以在任何时候设置STA，包括I²C在一个可寻址的从模式。

写“1”到 I2CONCLR 寄存器的STAC 位能清除STA位。当STA为0时，将不会产生START条件或重复START条件

在 STA和STO都被设置时，如果接口在主模式则在I²C总线上传输一个STOP条件，然后传输一个START 条件；如果I²C接口在从模式， 则会产生一个内部STOP 条件但不会在总线上传输。

STO是STOP标志位。设置这个位将导致I²C接口在主模式时传输一个 STOP条件，如果在从模式时则从错误条件中恢复。在主模式下，当STO为1时，在I²C总线上传输一个STOP条件。当总线检测到STOP条件时STO将被自动清除。

在从模式时，设置此位可以从错误条件中恢复。在这种情况下，将没有STOP条件传输到总线。硬件行为就像总线接收到了一个 STOP 条件一样,接口将转到“不可寻址地址”从接收模式。STO 标志位被硬件自动清除。

SI 是I²C中断标志位。 在I²C 状态变化时该位被设置。但是在进入F8状态时不会设置SI位，因为在那种情况下中断服务例程没有什么事情可做。

当 SI 被设置时，SCL 线上串行时钟的低电平时间被延长，串行传输暂停。当SCL为高电平时，它将不会影响SI 标志位。SI 必须由软件通过写1到I2CONCLR寄存器的SIC位来复位。

AA 是生效应答标志位。当被设置为1时，在 SCL 线上的应答时钟脉冲期间，如果有下面的情况下之一，将会返回一个应答信号 (SDA的低电平)：

1. 从地址寄存器的地址已经收到。
2. 当 I2ADR 中的广播位 (GC位) 被设置时，接收到广播地址。
3. 在 I²C为主接收模式时，接收到一个数据字节。
4. 在 I²C为可寻址的从接受模式时，接收到一个数据字节。

通过对I2CONCLR寄存器的AAC位写1可以清除AA位。当AA为0时，如果有下面的情况下之一，将会返回一个非应答信号 (SDA的高电平)

1. 在 I²C为主接收模式时，接收到一个数据字节。
2. 在 I²C为可寻址的从接受模式时，接收到一个数据字节。

7.2 I²C 状态寄存器 (I2CSTAT - 0x4000 0004)

每个 I²C 状态寄存器反映了相应 I²C接口的情况。I²C 状态寄存器只读。

表 140. I²C 状态 寄存器 (I2C0STAT - 0x4000 0004) 位域描述

位	符号	描述	复位值
2:0	-	这些位没有用到并且始终为 0。	0
7:3	Status	这些位给出了I ² C接口的实际状态信息。	0x1F
31:8	-	保留。从保留位读出的值未定义。	-

这三个最低有效位总是为0。作为一个字节，这个状态寄存器的内容表示一个状态码。一共有26种可能的状态码。当状态码为0xF8时，没有任何相关有用信息且不会对SI位置位。所有其他25个状态码与标准的I²C状态相符。当进入这些状态中的任何一个时，将对SI位置位。如需状态码的完整列表，请参阅表 10-155 到表 10-160。

7.3 I²C 数据寄存器 (I2C0DAT - 0x4000 0008)

这个寄存器包含将要发送的数据和刚接收的数据。当SI为1时，当该寄存器不处于移位过程时，CPU将读取和写入这个寄存器。只要SI位为1，I2DAT 寄存器中的数据就会保持稳定。I2DAT寄存器中的数据总是从右移到左：传送时，MSB (第7位)是被传送的第1位；接收时，收到数据的第1位位于I2DAT的 MSB 位。

表 141. I²C 数据寄存器 (I2C0DAT - 0x4000 0008) 位域描述

位	符号	描述	复位值
7:0	Data	这个寄存器保存要发出的数据或者已接收的数据。	0
31:8	-	保留。从保留位读出的值未定义。	-

7.4 I²C 从地址寄存器 0 (I2C0ADR0- 0x4000 000C)

该寄存器可读可写，且仅用在I²C接口设置为从模式时。主模式时，该寄存器无效。I2ADR的LSB位是广播位。当该位为1时，可识别广播地址 (0x00) 。

这些寄存器中，任何包含00x的寄存器将被禁止，且不会匹配总线上的任何地址。在复位时，将清除其的禁用状态。参见 表 10-148.

表 142. I²C 从地址寄存器 0 (I2C0ADR0- 0x4000 000C) 位域描述

位	符号	描述	复位值
0	GC	广播允许位。	0
7:1	Address	I ² C 从模式的设备地址。	0x00
31:8	-	保留。从保留位读出的值未定义。	-

7.5 I²C SCL HIGH 和 LOW 占空比寄存器(I2C0SCLH - 0x4000 0010 和 I2C0SCLL- 0x4000 0014)

表 143. I²C SCL HIGH 占空比寄存器 (I2C0SCLH - address 0x4000 0010) 位域描述

位	符号	描述	复位值
15:0	SCLH	SCL 高电平时间计数值。	0x0004
31:16	-	保留。从保留位读出的值未定义。	-

表 144. I²C SCL Low 占空比寄存器 (I2C0SCLL - 0x4000 0014) 位域描述

位	符号	描述	复位值
15:0	SCLL	SCL 低电平时间计数值。	0x0004
31:16	-	保留。从保留位读出的值未定义	-

7.5.1 选择适当的 I²C 数据率和占空比

必须通过软件设置寄存器I2SCLH和I2SCLL的值来选择适当的数据率和占空比。I2SCLH定义了SCL高电平持续时间的I2C_PCLK 周期数，I2SCLL定义了 SCL低电平持续时间的I2C_PCLK 周期数。频率由下式决定 (I2C_PCLK 是外围设备I2C的时钟频率):

(4)

$$I^2C_{bitfrequency} = \frac{I2CPCLK}{I2CSCLH + I2CSCLL}$$

I2SCLL和I2SCLH的值必须确保数据的传输速率在适合的I²C 数据速率范围内。每个寄存器的值必须大于或等于4。表 10-145 给了一些基于I2C_PCLK频率和 I2SCLL 和 I2SCLH 的值来确定 I²C-总线速率的例子。

表 145. 根据 I²C时为钟频率选择I2SCLL + I2SCLH值

I ² C mode	I ² C bit frequency	I ² C_PCLK (MHz)								
		6	8	10	12	16	20	30	40	50
		I ² SCLH + I ² SCLL								
标准模式	100 kHz	60	80	100	120	160	200	300	400	500
快速模式	400 kHz	15	20	25	30	40	50	75	100	125
增强型快速模式	1 MHz	-	8	10	12	16	20	30	40	50

I2SCLL 和 I2SCLH 没有必要是相同的。可以通过软件设置这两个寄存器的值来设置不同的 SCL占空比。I²C总线规范定义中，在快速模式和增强型快速模式时SCL 低电平时间和高电平时间就可不同。

7.6 I²C 控制清除寄存器(I2C0CONCLR - 0x4000 0018)

I2CONCLR寄存器控制I2CON 寄存器（用于控制 I²C接口）中位的清除。写入1到这个寄存器的某位，将导致I2C控制寄存器相应的位被清除。写一个0，则没有影响。

表 146. I²C 控制清除寄存器 (I2C0CONCLR - 0x4000 0018) 位域描述

位	符号	描述	复位值
1:0	-	保留，用户软件不要对保留位写‘1’。从保留位读取的值未定义。	NA
2	AAC	有效应答清除位	
3	SIC	I ² C 中断清除位。	0
4	-	保留，用户软件不要对保留位写‘1’。从保留位读取的值未定义。	NA
5	STAC	START 标志位清除位。	0
6	I2ENC	I ² C 接口禁用位。	0
7	-	保留，用户软件不要对保留位写‘1’。从保留位读取的值未定义。	NA
31:8	-	保留。从保留位读取的值未定义。	-

AAC是有效应答清除位。在I2CONSET寄存器中写1 将清除AA 位，写0无效。

SIC是I²C中断清除位。在I2CONSET寄存器中写1 将清除SI位，写0无效。

STAC 是START 标志清除位。在I2CONSET寄存器中写1 将清除STA位，写0无效。

I2ENC 是 I²C 接口禁用位。在I2CONSET寄存器中写1将清除I2EN位，写0无效。

7.7 I²C 监测模式控制寄存器 (I2COMMCTRL - 0x4000 001C)

该寄存器控制监督模式，监督模式允许I²C模块在其并不真正参与I²C 总线通讯的情况下，仍能检测I²C 总线上的通讯。

表 147. I²C 监测模式控制寄存器 (I2COMMCTRL0 - 0x4000 001C) 位域描述

位	符号	描述	复位值
0	MM_ENA	允许监测描述。	0
0		禁用监测模式。	
1		I ² C 模块将进入 监测模式。在该模式中 SDA 的输出将被强制为高电平。这将阻止I ² C 模块输出任何种类的数据(包括 ACK)到I ² C 数据总线上。 根据 ENA_SCL 位的状态，输出也可能被强制为高电平，以阻止模块对 I ² C 时钟线的控制。	

表 147. I²C 监测模式控制寄存器 (I2CMMCTRL0 - 0x4000 001C) 位域描述

位	符号	描述	复位值
1	ENA_SCL	SCL 输出允许。	0
		0 当该位被清为 ‘0’时， 在模块为监测模式时SCL 的输出将被强制为高电平。正如上面描述的， 阻止模块对 I ² C 时钟线的任何控制。	
		1 当此位被设置， I ² C模块对时钟线可行使与正常工作时相同的控制。这意味着， 作为一个从外设， I ² C模块可以“扩展”时钟线（保持低电平）， 直到它有时间响应一个I ² C中断。 ^[1]	
3	MATCH_ALL	选择中断寄存器匹配。	0
		0 当此为被清除， 只有当 一个地址与上述描述的4个地址寄存器之一匹配时， 才会产生一个中断。 也就是说， 只要识别出地址是相关的， 模块将作为一个正常的从模式响应。	
		1 当该位被置为 ‘1’ 并且I ² C 处于监测模式， 接收到任何地址都会产生一个中断。 这将使模块可监测总线上的所有通讯。	
31:4	-	- 保留。从保留位读取的值未定义。	

[1] 当 ENA_SCL 被清除并且 I²C不再有阻塞总线的能力时， 中断响应时间变得很重要。为了在这些情况下， 让模块有时间来响应一个I²C中断。 DATA_BUFFER寄存器(10-7.9节) 被用来保存接收的数据， 保存时间为9位字的传输时间。

注释: 如果MM_ENA为 ‘0’ (即不是监测模式)， ENA_SCL 和 MATCH_ALL位是无效的。

7.7.1 监测模式中的中断

在监测模式时， 模块所有的中断将如常产生。这意味着检测到一个地址匹配 (MATCH_ALL 为1时接收到任何地址， 否则就是接收到与四个地址寄存器其中任一相匹配的地址)将会产生第一个中断。

在一个地址匹配检测后， 中断将会在从设备写传输中的每个数据字节接收之后产生， 或在从设备读传输的每个字节发送成功之后产生。在第二种情况下， 数据寄存器实际包含了在总线上被其他从设备传输的数据， 这些从设备都被主设备寻址过。

跟着所有这些中断， 处理器可以读取数据寄存器， 看看总线上实际传输了什么。

7.7.2 监测模式下的仲裁丢失

在监测模式下， I²C将不能响应总线上主机发出的信息请求或者发出的一个 ACK。而总线上的一些其他的从设备将会响应。只要我们的模块是相关的， 这将极有可能导致仲裁状态丢失。

软件应该知道一个事实，即模块在监测模式下不会响应被检测到的任何仲裁状态丢失。此外，可以在模块中硬件实现，阻止一些或所有的仲裁状态（如果这些状态阻止了一个期望发生的中断或者产生一个不想要的中断）丢失。然而，是否添加这种硬件实现仍有待确定。

7.8 I²C 从地址寄存器 (I2C0ADR[1, 2, 3]- 0x4000 00[20, 24, 28])

该寄存器可读可写，且仅在I²C接口被设置为从模式时使用。在主模式时该寄存器无效。I2ADR 的LSB位是广播位。当该位为1时，广播地址 (0x00) 被识别。

这些寄存器中，任何包含位00x 的寄存器将被禁止，且不会与总线上的任何地址匹配。在复位时，四个从地址寄存器都将被清零到这个禁用状态

表 148. I²C 从地址寄存器 (I2C0ADR[1, 2, 3]- 0x4000 00[20, 24, 28]) 位域描述

位	符号	描述	复位值
0	GC	广播允许位。	0
7:1	Address	从模式时I ² C的设备地址。	0x00
31:8	-	保留。从保留位读取的值未定义。	0

7.9 I²C 数据缓冲寄存器 (I2C0DATA_BUFFER - 0x4000 002C)

在监测模式下，如果ENA_SCL 位为0，I²C 模块将没有扩展时钟(停止总线)的能力。这意味着该处理器将必须在一个有限的时间之内去读取总线上收到的数据的内容。如果处理器和平常一样读取I2DAT移位寄存器，那么在接受数据被新的数据覆盖之前，它只能有一个位的时间来响应中断。

为了让处理器有更多的响应时间，添加一个新的8位、只读寄存器DATA_BUFFER。在每次从总线上接收9位数据（8位加ACK或NACK）后，将I2DAT 的前八位的内容传送到DATA_BUFFER寄存器。这意味着处理器在数据被覆盖之前将有9位传输时间响应中断并读取数据。

像平时一样，处理器将仍然有能力直接读取I2DAT，且I2DAT的行为将不会有任何方式改变。

虽然DATA_BUFFER寄存器主要在监测模式和ENA_SCL位= '0'时使用，但是它可在任何运行模式下的任何时间都可以被读取。

表 149. I²C 数据缓冲寄存器 (I2CDATA_BUFFER - 0x4000 002C)位域描述

位	符号	描述	复位值
7:0	Data	该寄存器保存I2DAT移位寄存器前8位（8 MSBs）的值	0
31:8	-	保留。从保留位读取的值未定义。	0

7.10 I²C 屏蔽寄存器 (I2C0MASK[0, 1, 2, 3] - 0x4000 00[30, 34, 38, 3C])

四个屏蔽寄存器每个包含七个有效位 (7:1)。若些寄存器中任何一位设置为 ‘1’ 时，当接收到的地址和I2ADDRn寄存器相比较时，将会导致接收到的地址的相应位和那个屏蔽寄存器做自动比较。换句话说，I2ADDRn 寄存器中被屏蔽的位将不会参与决定地址的匹配。

在复位时，所有的屏蔽寄存器都清 ‘0’。

屏蔽寄存器在比较广播地址 (“0000000”)时没有用。

屏蔽寄存器的位(31:8)位(0) 没有使用，且不能写入。读这些位总是返回0。

当地址匹配中断发生时，处理器将读取数据寄存器（I2DAT），以确定实际上是那个地址导致了地址匹配。

表 150. I²C 屏蔽寄存器 (I2C0MASK[0, 1, 2, 3] - 0x4000 00[30, 34, 38, 3C]) 位域描述

位	符号	描述	复位值
0	-	保留，用户软件不要对保留位写‘1’。读取这些位总是返回‘0’。	0
7:1	MASK	屏蔽位。	0x00
31:8	-	保留。从保留位读取的值未定义。	0

8. I²C 操作模式

在一个给定的应用程序中，I²C模块可作为一个主设备，一个从设备，或两者兼有。在从模式时，I²C硬件将监听总线，是否有四个从设备地址之一和广播地址， 如果找到其中的任何一个就会发出中断请求。如果处理器希望成为总线主机， 在进入主模式之前必须等待直到总线空闲，这样可以保证一个可能正在进行的从操作不会被中断。如果在主模式时总线仲裁丢失， I²C 模块会立即切换到从模式，并可以在同一个串行传输中检测到其自己的从设备地址。

8.1 主发送器模式

在这种模式下，数据从主设备传输从设备。在可以进入主发送模式之前，必须如表 10-151所示初始化I2CONSET寄存器。I2EN 必须设置为1，以允许I²C 功能。如果AA 位为0，当其他设备是总线主机时，I²C接口将不识别任何地址，所以它不能进入从模式。STA、STO和SI位必须为0，其中SI 位可以通过写‘1’到I2CONCLR寄存器的SIC位清除；STA 位必须在写从地址之后清除。

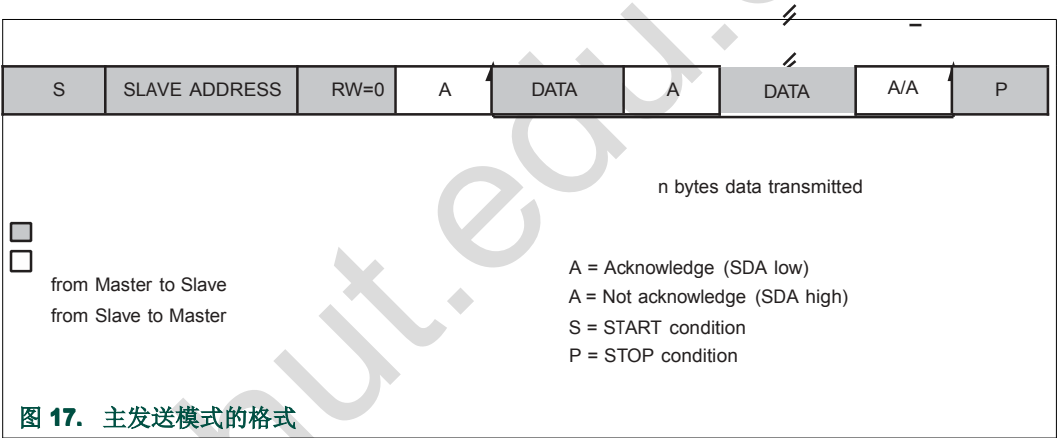
表 151. 用于配置主模式的 I2C0CONSET 和 I2C1CONSET

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	0	-	-

传送的第一个字节包含接收设备的从地址（7位）和数据方向位。在此模式下，数据方向位（读/写）应该是0，这意味着写入。传送的第一个字节包含接收设备的从地址和数据写入位。每次传输8位数据，在每个字节传输完成后会接收到一个应答位。START条件和STOP条件的输出表明一个串行传输的开始和结束。

当软件置位STA位时，I²C接口将进入主发送模式。总线一旦空闲，I²C逻辑将发送START条件；START条件发送后，SI位将被置1；I2STAT寄存器中的状态码是0x08，这个状态码将会作为一个状态服务程序的向量；状态服务程序将会加载从设备地址和写I2DAT寄存器，并清除SI位。SI位可以通过写‘1’到I2CONCLR寄存器的SIC位清除。

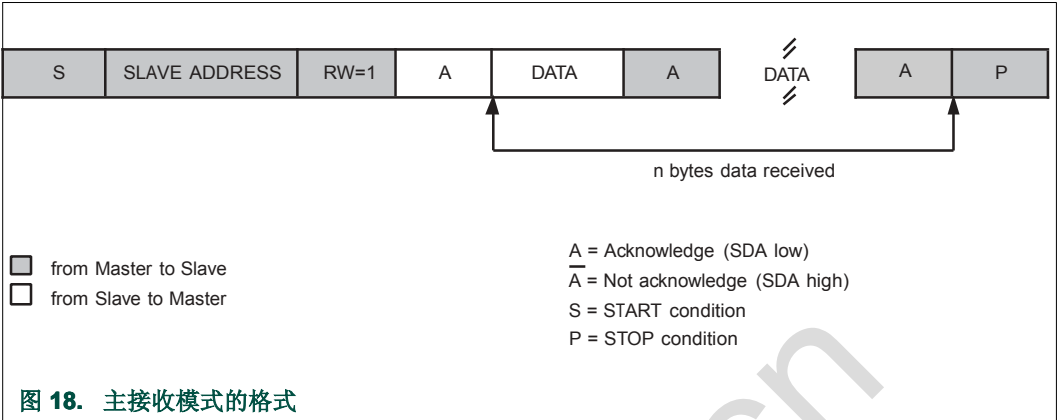
当从地址和读/写位已经发送并且也收到了应答位，SI位将会再次被置1，对于主模式现在可能的状态码是0x18、0x20或0x38，而对于从模式允许(通过设置AA为1)时则可能是0x68、0x78或者0xB0。这些状态码中的每一个状态都会有一个相应的动作，具体动作如表 10-155 到表 10-160所示。



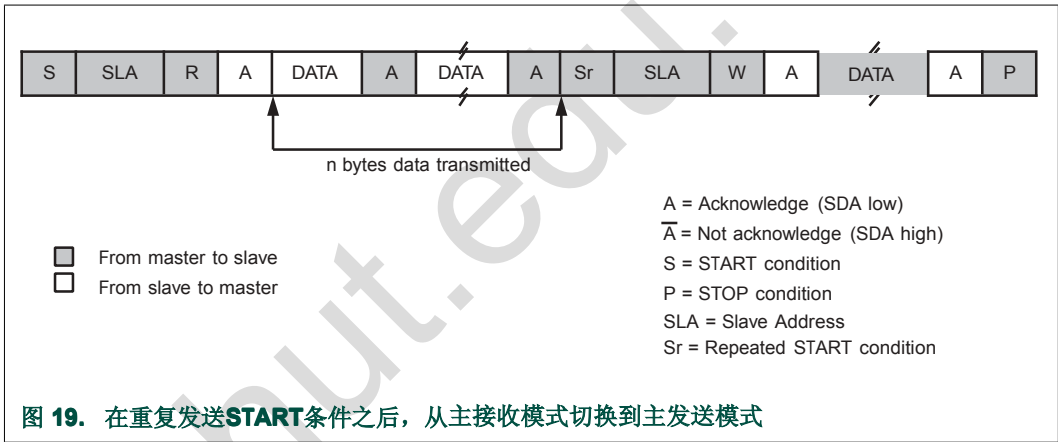
8.2 主接收模式

在主接收模式，从一个从发送器接收数据。传输以与主发送模式同样的方式启动。当START条件已经传送，中断服务程序必须加载从地址和数据方向位到I²C数据寄存器（I2DAT）中，然后清除的SI位。在这种情况下，数据方向位（读/写）应为1，表示读。

当从地址和数据方向位已发送，且已收到一个应答位时，SI位也被设置为1，状态寄存器将显示状态代码。对于主模式，可能的状态代码是0x40、0x48或0x38；对于从模式，可能的状态代码是0x68、0x78或0xB0。有关详情，请参阅表 10-156。



在一个重复的起始条件后，I²C 会转到主发送模式。



8.3 从接收模式

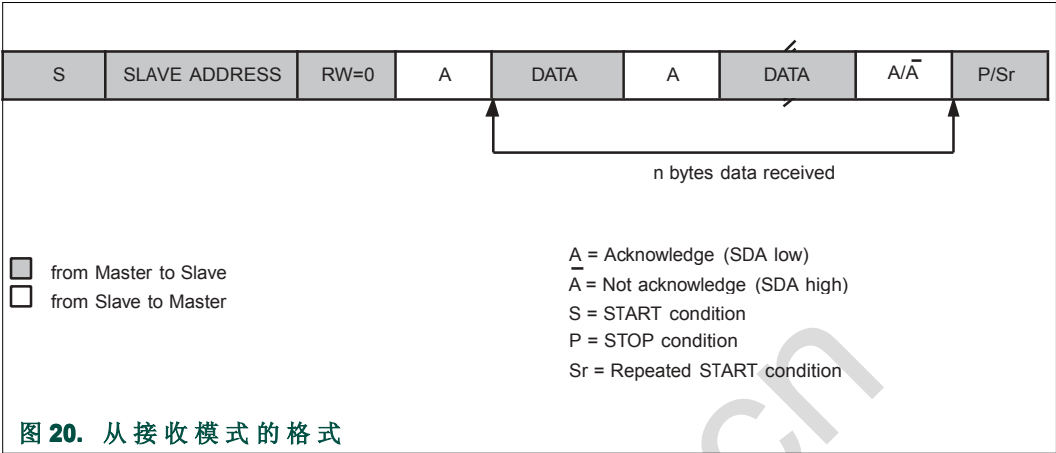
在从接收模式中，从一个主发送器接收数据字节。为了初始化从接收器模式，需要对任意一个地址寄存器（I2ADR0 - 3）写从地址，还要按表10-152所示去写I2C控制设置寄存器（I2CONSET）。

表 152. 用于配置从模式的I2C0CONSET和 I2C1CONSET

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	1	-	-

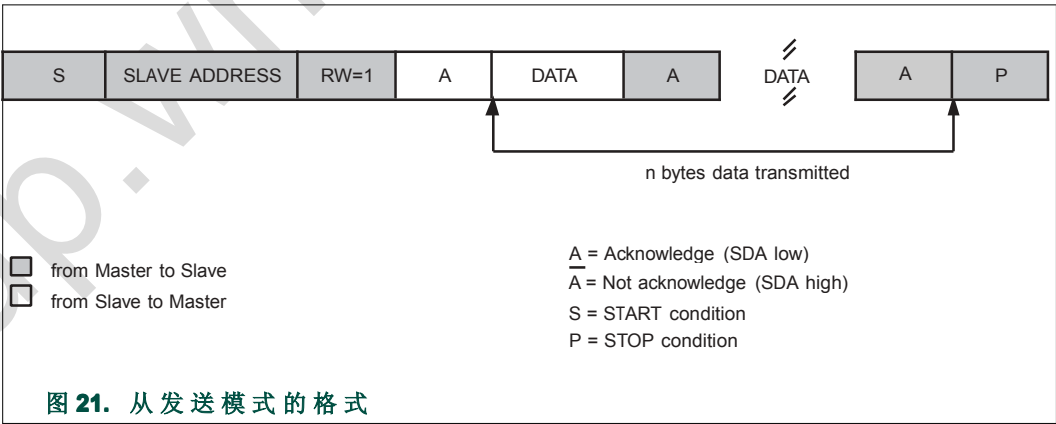
I2EN位必须设置为1，以允许I²C的功能。AA位必须设置为1，以识别自己从地址或广播地址。STA、， STO和SI位均要设置为0

在 I2ADR 和 I2CONSET被初始化之后，I²C接口将等待接收其自身地址或广播地址，地址后面带有数据方向位。如果方向位0（写），它进入从接收器模式；如果方向位为1（读），则进入从发射模式。在从地址和方向位均已收到之后，SI位将设置为1，可以从状态寄存器（I2STAT）中读取有效状态代码。状态码和相应动作可参考表10-159。



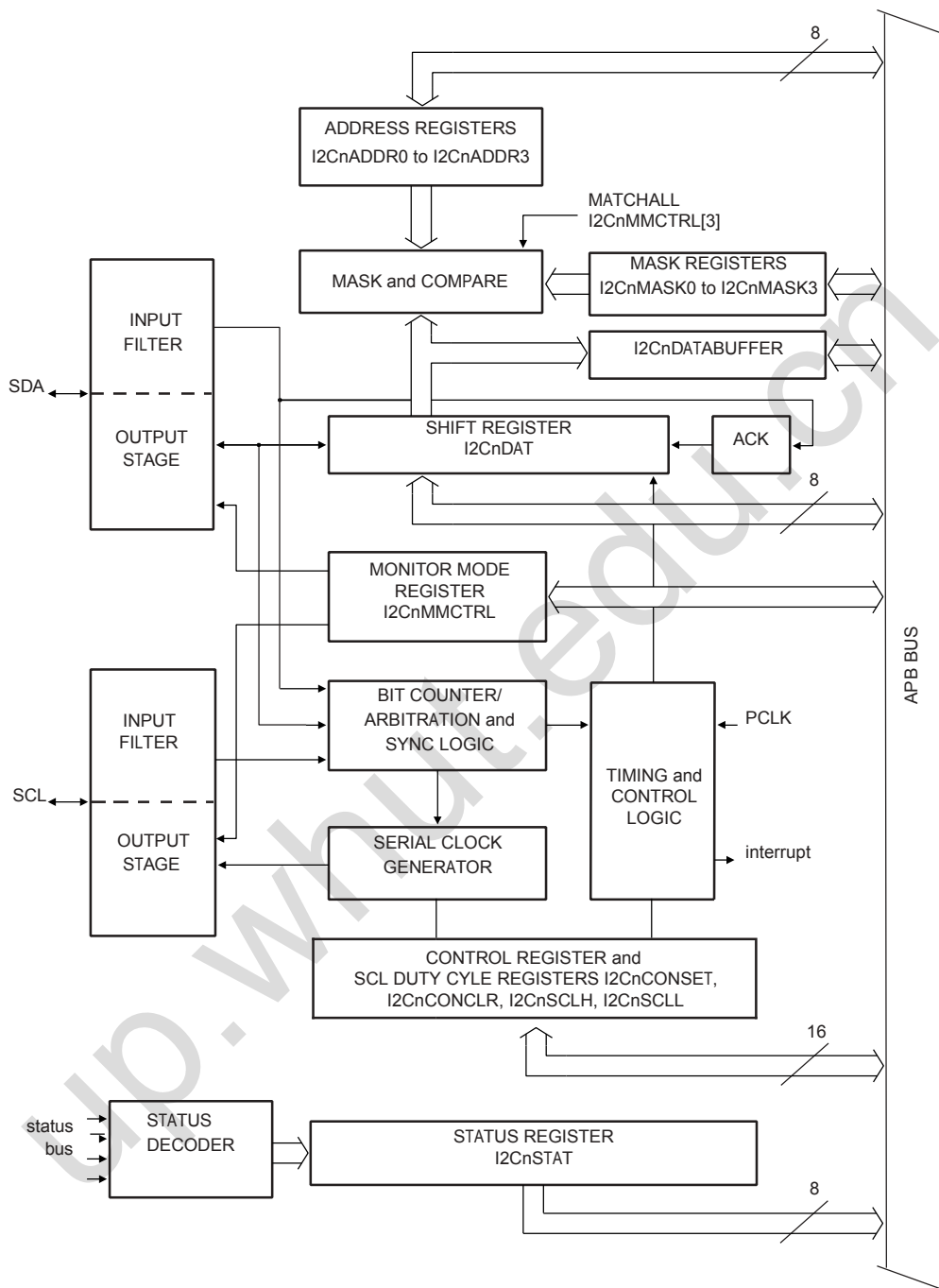
8.4 从发送模式

第一个字节的接收和处理与从接收模式一样，但该模式下方向位为1，表示读操作。串行数据通过SDA传输，串行时钟由SCL输入。START条件和STOP条件用于识别一个串行传输的开始和结束。在一个给定的应用程序中，I²C可用于主模式或从模式。在从模式中，I²C硬件寻找其自己的从地址和广播地址。如果检测到这些地址之一，将会产生一个中断请求。如果处理器希望成为总线主机，那么在进入主模式之前必须等待直到总线空闲，这样可能正在进行的从操作不会被中断。如果在主模式下总线仲裁被丢失，I²C接口会立即切换到从模式，并可以在同一个串行传输中检测到其自己的从设备地址。



9. I²C 的实现与操作

图 10-22 所示为如何片上实现 I²C总线接口的，其中每个模块的功能在后面介绍。

图 22. I²C 串行接口方框图

9.1 输入滤波器 (Input filter) 和输出段 (output stage)

输入信号与内部时钟同步, 少于3时钟周期的毛刺将被过滤掉。

为了符合I²C规范, I²C的输出设计为一个特殊的引脚。

9.2 地址寄存器, I2ADDR0 - I2ADDR3

这些寄存器可装载7位从地址（最高7位），在I²C模块被编程配置为从发送或从接收模式时，会对这些地址做出相应。最低位(GC)被用来允许广播地址(0x00)被识别。允许多个从地址时，如果已收自身从地址，可以从I2DAT寄存器中读出实际接收到的地址。

9.3 地址屏蔽寄存器, I2MASK0 - I2MASK3

四个屏蔽寄存器每个包含七个有效位(7:1)。若些寄存器中任何一位设置为‘1’时，当接收到的地址和I2ADDRn寄存器相比较时，将会导致接收到的地址的相应位和那个屏蔽寄存器做自动比较。换句话说，I2ADDRn寄存器中被屏蔽的位将不会参与决定地址的匹配。

如果I2ADDRn第0位(GC允许位)为1并且位(7:1)都为0，那么模块将会响应收到的地址“0000000”，而不管相关屏蔽寄存器的状态。

当地址匹配中断发生时，处理器将读取数据寄存器(I2DAT)，以确定实际上是哪个地址导致了地址匹配

9.4 比较器

比较器会将收到的7位从设备地址与自身的从设备地址(I2ADR中最高7位)比较。它也会将首先收到的8位与广播地址(0x00)进行比较。一旦发现相等的情况，就会设置一个合适的状态码并产生一个中断请求。

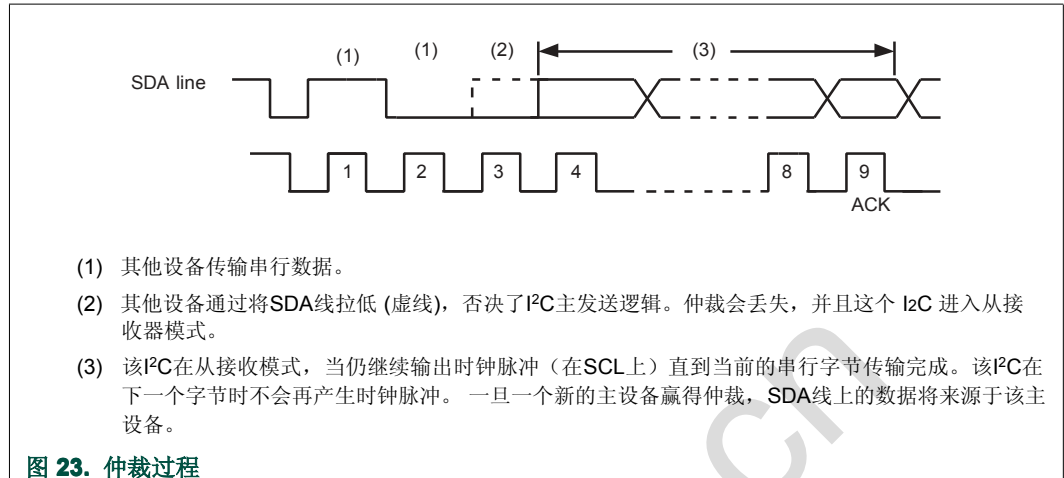
9.5 移位寄存器, I2DAT

这个8位寄存器包含了一个字节的串行数据，该数据是即将发送的或是刚接收的。I2DAT的数据总是从右移到左，第一个被发出的位是最高位（第7位），在一个字节已经接收完成后，最先接收到的数据放在I2DAT的最高位。当数据移出的同时，总线上的数据被移入。I2DAT总是保存总线上最后一个字节的数据。因此，在仲裁丢失的情况下，主发送器会将I2DAT中正确的数据传输到从接收器中。

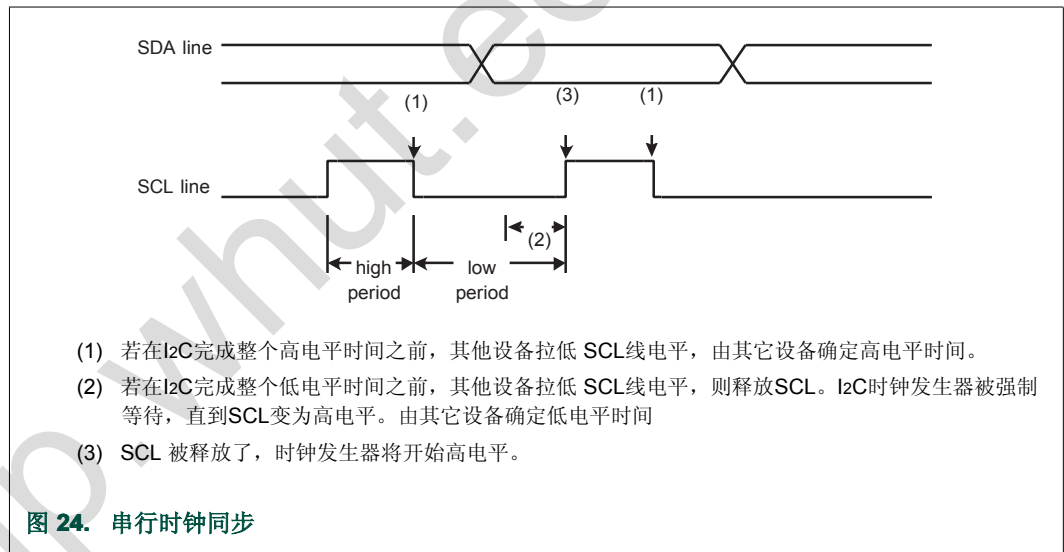
9.6 仲裁和同步逻辑

在主控发送模式下，仲裁逻辑检查每个传输逻辑1，也就是实际在I²C总线出现的逻辑1。如果总线上的另一个设备否决了逻辑1，把SDA拉低，仲裁丢失，而且I²C模块立即从主发送变为从接收。I²C模块将继续输出时钟脉冲（在SCL上）直到当前的串行字节传输完成。

在主接收模式下，仲裁也可能丢失。在此模式下，只可有I²C模块返回一个“非应答”（逻辑1）到总线上时才发生仲裁丢失。当总线上其他设备拉低这个信号时仲裁丢失。因为这只可能发生在一个串行字节结束时，故I²C块没有不再产生的时钟脉冲。[图 10-23](#)所示为仲裁过程。



同步逻辑将串行时钟发生器与其他设备SCL线的时钟脉冲同步。如果两个或两个以上的主设备产生时钟脉冲，高电平的持续时间取决于设备生成最短高电平，低电平持续时间则取决于设备生成最长的低电平。图 10-24 为同步过程。



从设备可以拉伸低电平时间以将总线主机速度放慢。另外，当为了实现握手，低电平时间也可能被拉伸。这可以在每个位或一个完整的字节传输之后发生。在完成一个字节发送或完成一个字节接收且传输一个应答位之后，I²C 模块将拉伸 SCL 的低电平时间。串行中断标志 (SI) 被设置为1，低电平时间将拉伸，直到串行中断标志被清除。

9.7 串行时钟发生器

当I²C模块在主发送或主接收模式时，这种可编程时钟脉冲发生器提供SCL时钟脉冲。当I²C 模块在从模式时，其将关闭。通过I²C时钟控制寄存器，设置I²C输出时钟频率和占空

比。详见I2CSCLL和I2CSCLH寄存器描述。输出时钟脉冲按编程设置产生占空比，除非该总线与其他SCL时钟源同步（如上所述）。

9.8 定时与控制

定时和控制逻辑产生用于串行字节处理的定时和控制信号。该逻辑模块为I2DAT提供移位脉冲，允许比较器，产生和检测START和STOP条件，接收和发送应答位，控制主从模式，包含了中断请求逻辑，还监测I2C总线状态。

9.9 控制寄存器，I2CONSET 和 I2CONCLR

I2C 控制寄存器位用于控制I2C模块的以下功能：启动和重新启动串行传输、终止串行传输、比特率、地址识别和确认。

I2C 控制寄存器的内容可通过I2CONSET读出。对 I2CONSET寄存器相应位写1，将会设置I2C 寄存器的相应位。对I2CONCLR寄存器相应位写1，则将会清除 I2C 寄存器的相应位。

9.10 状态解码器和状态寄存器

状态解码器获取所有内部状态位，并将它们压缩成了一个5位的代码。每个I2C总线状态都有唯一的代码。5位的编码，可用于生成各种服务程序快速处理的向量地址。每个服务程序处理一个特定总线状态。如果I2C使用了所有的四种模式，共有26种可能的总线状态。在串行中断标志被硬件设置时，5位的状态代码锁存到状态寄存器的最高5位中，并保持稳定直到中断标志被软件清除。这个状态寄存器的最低3位始终为零。如果状态码用作服务程序的向量地址，则程序被8个地址位置代替。对大多数服务程序来说，8个字节的代码是足够的（见本节软件举例）。

10. I2C 操作模式详细介绍

四个操作模式:

- 主发送
- 主接收
- 从接收
- 从发送

每种模式的数据传输如 图 10-25、图 10-26、图 10-27、图 10-28和图 10-29。表 10-153 列出了这些图在描述I2C操作模式时所用到的缩写。

表 153. 描述 I²C 操作时用到的缩写

缩写	说明
S	START 条件
SLA	7 位地址
R	读位 (SDA上高电平)
W	写位 (SDA上低电平)
A	应答位 (SDA上低电平)
\overline{A}	非应答位 (SDA上高电平)
Data	8位数据字节
P	STOP 条件

在图 10-25 到 图 10-29中, 圆圈用于指示当串行中断标志被设置。圆圈中的数字为保存在 I2STAT寄存器中的状态代码。在这些点上, 服务程序必须执行以继续或完成串行传输。这些服务例程并不紧急, 因为直至串行中断标志由软件清除之前串行传输是被挂起的。

当进入一个串行中断程序时, I2STAT中的状态代码用于跳转到相应的服务程序中。对于每一个状态代码, 所需软件动作和下一个串行传输的详情见表 10-155到表 10-161。

10.1 主发送模式

在主控发送模式, 大量的数据字节数被传送到一个从接收器(见图10-25)。在进入主发送器模式之前, I2CON 必须做如下初始化:

表 154. I2CONSET 用于初始化主发送器模式

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	x	-	-

必须通过I2SCLL和 I2SCLH寄存器配置I²C 速率。I2EN 必须设置为逻辑1, 以启用I²C模块。如果AA 位被复位, I²C模块将不响应其自身从地址或广播地址, 即使其他设备成为总线主机。换句话说, 当 AA 复位时, I²C接口不能进入从模式, STA、STO和 SI 必须都复位。

现在可通过设置STA位进入主发送模式。现在，I²C逻辑将会测试I²C总线，一旦总线空闲将生成一个START条件。当一个START条件被传输，串行中断标志位（SI）被设置，且状态寄存器（I2STAT）的状态代码将是0x08。中断服务程序使用此状态代码，以进入一个正确的状态服务程序，该状态服务程序将加载从机地址和数据方向位（SLA+W）到I2DAT。在串行传输继续之前，I2CON中的SI位必须被复位。

当从地址和方向位已传输，且已收到一个应答信号，串行中断标志（位SI）将再次被设置，I2STAT中的状态码有多种可能。主模式下可能有0x18、0x20或0x38；如果从模式被允许（AA = 逻辑1）则可能有0x68、0x78或0xB0。这些状态代码所对应的动作详见表10-155。经过在重复START条件之后（状态0x10），I²C模块将通过加载SLA+R到I2DAT以转换到主接收模式

表 155. 主发送模式

状态码 (I2CSTAT)	I2C 和硬件总线状态	应用软件响应 To/From I2DAT	I2C硬件的下一个动作			
			To I2CON		AA	
			STA	STO		
0x08	已发送START 条件	加载 SLA+W; 清除 STA	X	0	0	X 将传输SLA+W; 将会接收到ACK 位。
0x10	已发重复START 条件	加载 SLA+W 或	X	0	0	X 同上。
		加载 SLA+R; 清除 STA	X	0	0	X 将传输SLA+W; I2C模块转换到 MST/REC模式。
0x18	已传输SLA+W; 已收到了ACK。	加载数据字节或	0	0	0	X 将传输数据字节; 将接收到ACK 位。
		无 I2DAT 动作或	1	0	0	X 传输重复START条件。
		无 I2DAT 动作或	0	1	0	X 传输STOP条件; STO标志将复位。
		无 I2DAT 动作	1	1	0	X 将传输STOP 条件之后传输一个 START 条件; STO 标志将被复位。
0x20	已传输SLA+W; 已收到了ACK。	加载数据字节或	0	0	0	X 将传输数据字节; 将会接收到ACK 位。
		无 I2DAT 动作或	1	0	0	X 传输重复START条件。
		无 I2DAT 动作或	0	1	0	X 将传输STOP条件; STO标志将复位。
		无 I2DAT 动作	1	1	0	X 将传输STOP 条件, 之后再传输一个 START 条件; STO 标志将被复位。
0x28	I2DAT中的数据字节已 传输, 已收到 ACK 。	加载数据字节或	0	0	0	X 将传输数据字节; 将会接收到ACK 位。
		无 I2DAT 动作或	1	0	0	X 传输重复START条件。
		无 I2DAT 动作或	0	1	0	X 将传输STOP条件; STO标志将复位。
		无 I2DAT 动作	1	1	0	X 将传输STOP 条件, 之后再传输一个 START 条件; STO 标志将被复位。
0x30	I2DAT中的数据字节已 传输, 已收到 NOT ACK	加载数据字节或	0	0	0	X 将传输数据字节; 将会接收到 ACK 位。
		无 I2DAT 动作或	1	0	0	X 传输重复START条件。
		无 I2DAT 动作或	0	1	0	X 将传输STOP条件; STO标志将复位。
		无 I2DAT 动作	1	1	0	X 将传输STOP 条件之后传输一个 START 条件; STO 标志将被复位。
0x38	在数据字节或 SLA+R/W传输阶 段, 仲裁丢失了。	无 I2DAT 动作或	0	0	0	X 将释放I2C总线, 将进入不可寻址从模 式。
		无 I2DAT 动作	1	0	0	X 当总线空闲时, 将发生START 条件。

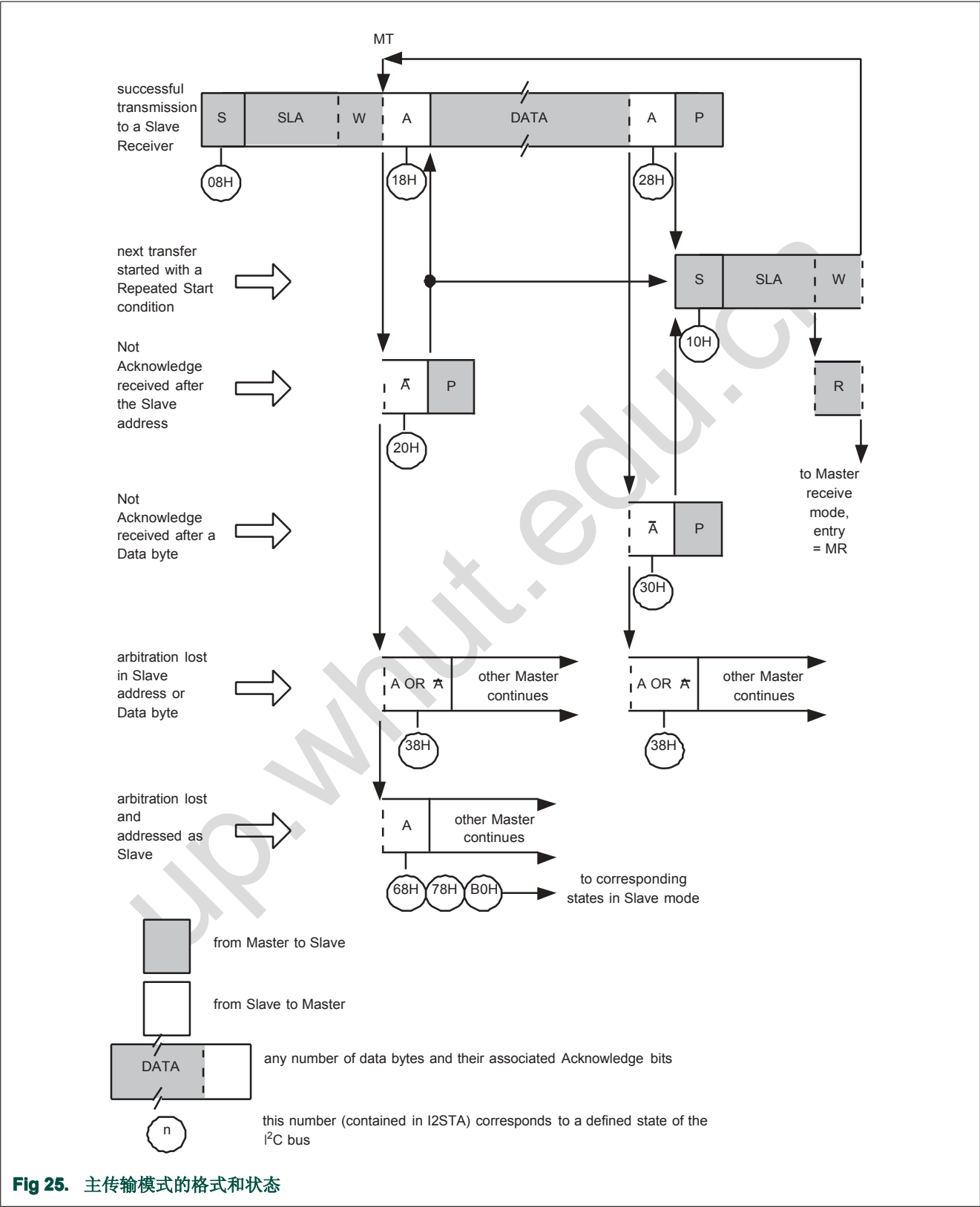


Fig 25. 主传输模式的格式和状态

10.2 主接收模式

在主接收模式中，从从发送器接收大量的数据字节(见 图 10-26)。传输被初始化为主发射模式。当START条件已传输时，中断服务程序必须加载7位从地址和数据方向位(SLA+R)到I2DAT。在串行传输继续之前，I2CON中的SI位必须被清零。

当已完成从地址和方向位传输，且已收到一个应答信号时，串行中断标志（位SI）将被再次设置，且I2STAT中的状态码有多种可能。主模式可能有 0x40、0x20或0x38；如果从模式被允许 (AA = 逻辑 1)则可能有0x68、0x78或0xB0。这些状态代码所对应的动作详见表 10-156。经过重复START 条件之后（状态0x10）。I²C模块 块将通过加载SLA+W 到I2DAT以转换到主发送器模式。

表 156. 主接收模式

状态码 (I2CSTAT)	I2C总线及硬件状态	应用软件响应 To/From I2DAT	To I2CON			AA	I2C硬件的下一个动作
			STA	STO			
0x08	已完成START 条件传输。	加载 SLA+R	X	0	0	X	将发送SLA+R，将收到ACK位
		加载 SLA+R 或	X	0	0	X	同上。
0x10	已完成一个重复 START 条件传输。	加载 SLA+W	X	0	0	X	将发送SLA+W，I2C模块将切换到 MST/TRX模式。
0x38	在NOT ACK位传输阶段，仲裁丢失	无I2DAT动作 或	0	0	0	X	I2C总线将被释放，I2C模块将进入从模式。
		无I2DAT动作	1	0	0	X	当总线空闲时，将发送START条件。
0x40	SLA+R已传输; 已收到 ACK	无I2DAT动作 或	0	0	0	0	将收到数据字节; 将返回NOT ACK位。
		无I2DAT动作	0	0	0	1	将收到数据字节; 将返回NOT ACK位。
0x48	SLA+R已传输; 已收到 NOT ACK	无I2DAT动作 或	1	0	0	X	将发送START条件。
		无I2DAT动作 或	0	1	0	X	将发送STOP条件， STO标志将复位。
		无I2DAT动作	1	1	0	X	将发送STOP 条件，之后再传输一个 START 条件; STO 标志将被复位。
0x50	已收到数据字节; 已返回ACK。	读数据字节 或	0	0	0	0	将收到数据字节; 将返回NOT ACK位。
		读数据字节	0	0	0	1	将收到数据字节; 将返回ACK位。
		读数据字节 或 1	1	0	0	X	将发送重复START条件。
0x58	已收到数据字节; 已返回NOT ACK	读数据字节 或 0	0	1	0	X	将发生STOP条件; STO 标志将被复位。
		读数据字节	1	1	0	X	将发送STOP 条件，之后再传输一个 START 条件; STO 标志将被复位。

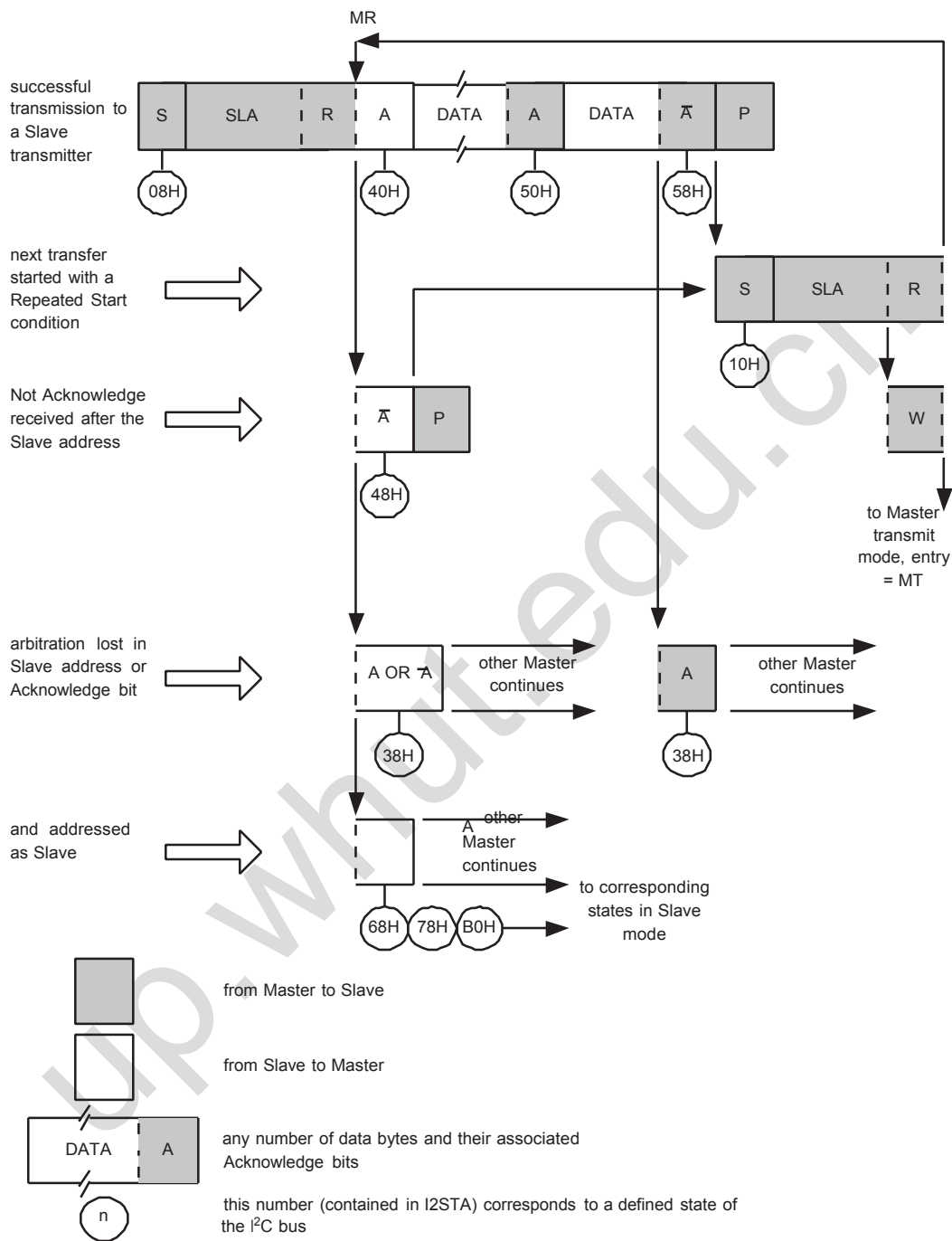


图26. 主接收模式的格式与状态

10.3 从接收模式

在从接收模式下，从主发送器接收数据字节 (见 图 10-27)。为了初始化从接收模式，I2ADR 和 I2CON 必须如下加载：

表 157. 从接收模式中I2C0ADR 和 I2C1ADR的用法

位	7	6	5	4	3	2	1	0
符号	自身 7-位从地址							GC

前面的 7 是当主设备发出地址时，I²C模块将响应的地址。如果设置了LSB (GC)，I²C模块将会响应广播地址 (0x00)； 否则它将忽略广播地址。

表 158. 用于初始化从接收模式的I2C0CONSET 和 I2C1CONSET

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	1	-	-

在从模式时，I²C总线速度的设置不会影响到I²C模块。I2EN 必须设置为逻辑 1 以允许I²C模块。AA 位必须设置为1，以使I²C 块能够应答自身的从地址或广播地址。STA、STO和SI 都必须复位。

当I2ADR和I2CON已被初始化之后，I²C模块将等待，直到它收到与其从地址相同的地址，并且如果I²C模块要工作在从接收模式下，则随后的数据方向位必须是“0”（W）。在已收到自身从机地址和W位后，串行中断标志（SI）被设置且可以从I2STAT读取有效状态代码。每个状态代码都对应合适的动作，具体动作详见表 10-159。如果I²C模块在主模式时仲裁丢失，也可以进入从接收模式 (见 状态码 0x68 和 0x78)。

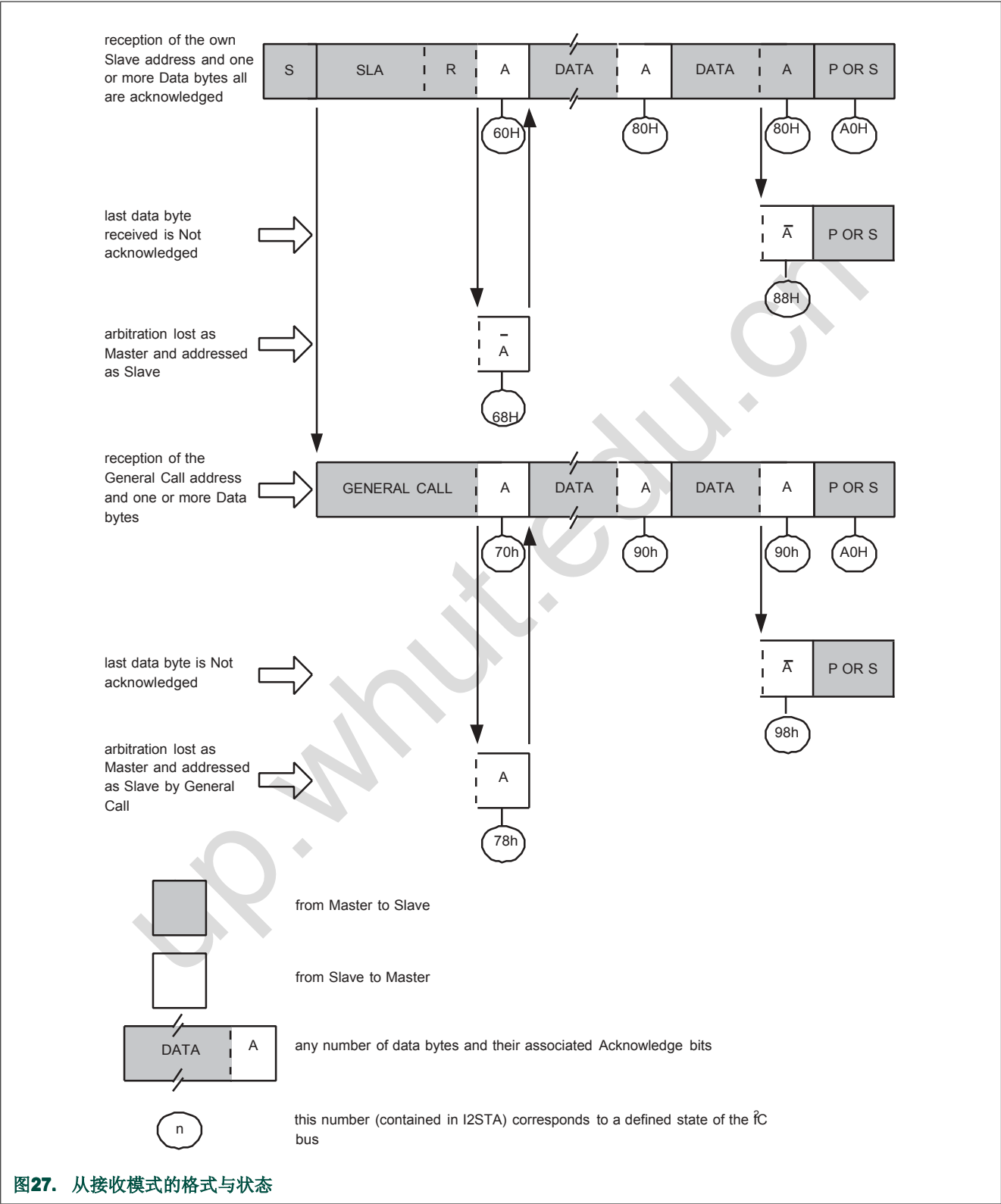
如果在传输过程中AA位被复位，I²C模块在接收到下一个数据字节之后，将返回一个非应答（逻辑1）到SDA线上。当AA复位时，I²C块将不会响应自己的从地址或广播地址。然而，I²C总线仍然被监测，并可在任何时间设置AA恢复地址识别。这意味着，AA位可用于暂将I²C模块与I²C总线隔离。

表 159. 从接收模式

状态码 (I2CSTAT)	I2C总线及硬件状态	应用软件响应 To/From I2DAT	I2C硬件的下一个动作				
			To I2CON		AA		
			STA	STO			
0x60	已经收到自身SLA+W； 已返回ACK。	无I2DAT动作 或	X	0	0	0	将接收数据字节，且将返回NOT ACK。
		无I2DAT动作	X	0	0	1	将接收数据字节，且将返回 ACK。
0x68	作为主控器，在SLA+R/W传输阶段，仲裁丢失；已收到自身SLA+W，返回ACK。	无I2DAT动作 或	X	0	0	0	将接收数据字节，且将返回NOT ACK。
		无I2DAT动作	X	0	0	1	将接收数据字节，且将返回 ACK。
0x70	已收到广播地址(0x00) 已返回ACK	无I2DAT动作 或	X	0	0	0	将接收数据字节，且将返回NOT ACK。
		无I2DAT动作	X	0	0	1	将接收数据字节，且将返回 ACK。
0x78	作为主控器，在SLA+R/W传输阶段，仲裁丢失； 已收到广播地址，返回ACK。	无I2DAT动作 或	X	0	0	0	将接收数据字节，且将返回NOT ACK。
		无I2DAT动作	X	0	0	1	将接收数据字节，且将返回 ACK。
0x80	之前用自身SLV地址寻址；已收到DATA；已返回ACK。	读数据字节 或	X	0	0	0	将接收数据字节，且将返回NOT ACK。
		读数据字节	X	0	0	1	将接收数据字节，且将返回 ACK。
0x88	之前用自身SLA地址寻址；已收到DATA；已返回NOT ACK。	读数据字节 或 0		0	0	0	转换到不可寻址 SLV模式；不能识别自身SLA 地址或广播地址。
		读数据字节 或 0		0	0	1	转换到不可寻址 SLV模式；将识别自身SLA 地址，如果I2ADR[0] = 1则识别广播地址。
		读数据字节 或 1		0	0	0	转换到不可寻址 SLV模式；不能识别自身SLA 地址或广播地址。当总线空闲时，将发送一个START条件。
		读数据字节	1	0	0	1	转换到不可寻址 SLV模式；将识别自身SLA 地址，如果I2ADR[0] = 1则识别广播地址。当总线空闲时，将发送一个START条件。
0x90	之前用广播地址寻址；已收到DATA；已返回ACK。	读数据字节 或 X		0	0	0	将接收数据字节，且将返回NOT ACK。
		读数据字节	X	0	0	1	将接收数据字节，且将返回 ACK。

表 159. 从接收模式 ...续

状态码 (I2CSTAT)	I2C总线及硬件状态	应用软件响应		I2C硬件的下一个动作			
		To/From I2DAT	To I2CON		AA		
			STA	STO			
0x98	之前用广播地址寻址; 已收到DATA; 已返回 NOT ACK。	读数据字节 或 SLA	0	0	0	0	转换到不可寻址 SLV模式; 不能识别自身地址或广播地址。
		读数据字节 或	0	0	0	1	转换到不可寻址 SLV模式; 将识别自身SLA地址, 如果I2ADR[0] = 1则识别广播地址。
		读数据字节 或	1	0	0	0	转换到不可寻址 SLV模式; 不能识别自身SLA地址。当总线空闲时, 将发送一个START条件。
		读数据字节	1	0	0	1	转换到不可寻址 SLV模式; 将识别自身SLA地址, 如果I2ADR[0] = 1则识别广播地址。当总线空闲时, 将发送一个START条件。
0xA0	当作为SLV/REC 或 SLV/TRX寻址时, 已收到 STOP条件或 重复START条件。	无STDAT动作 或 SLA	0	0	0	0	转换到不可寻址 SLV模式; 不能识别自身地址或广播地址。
		无STDAT动作 或	0	0	0	1	转换到不可寻址 SLV模式; 将识别自身SLA地址, 如果I2ADR[0] = 1则识别广播地址。
		无STDAT动作 或	1	0	0	0	转换到不可寻址 SLV模式; 不能识别自身SLA地址。当总线空闲时, 将发送一个START条件。
		无STDAT动作	1	0	0	1	转换到不可寻址 SLV模式; 将识别自身SLA地址, 如果I2ADR[0] = 1则识别广播地址。当总线空闲时, 将发送一个START条件。



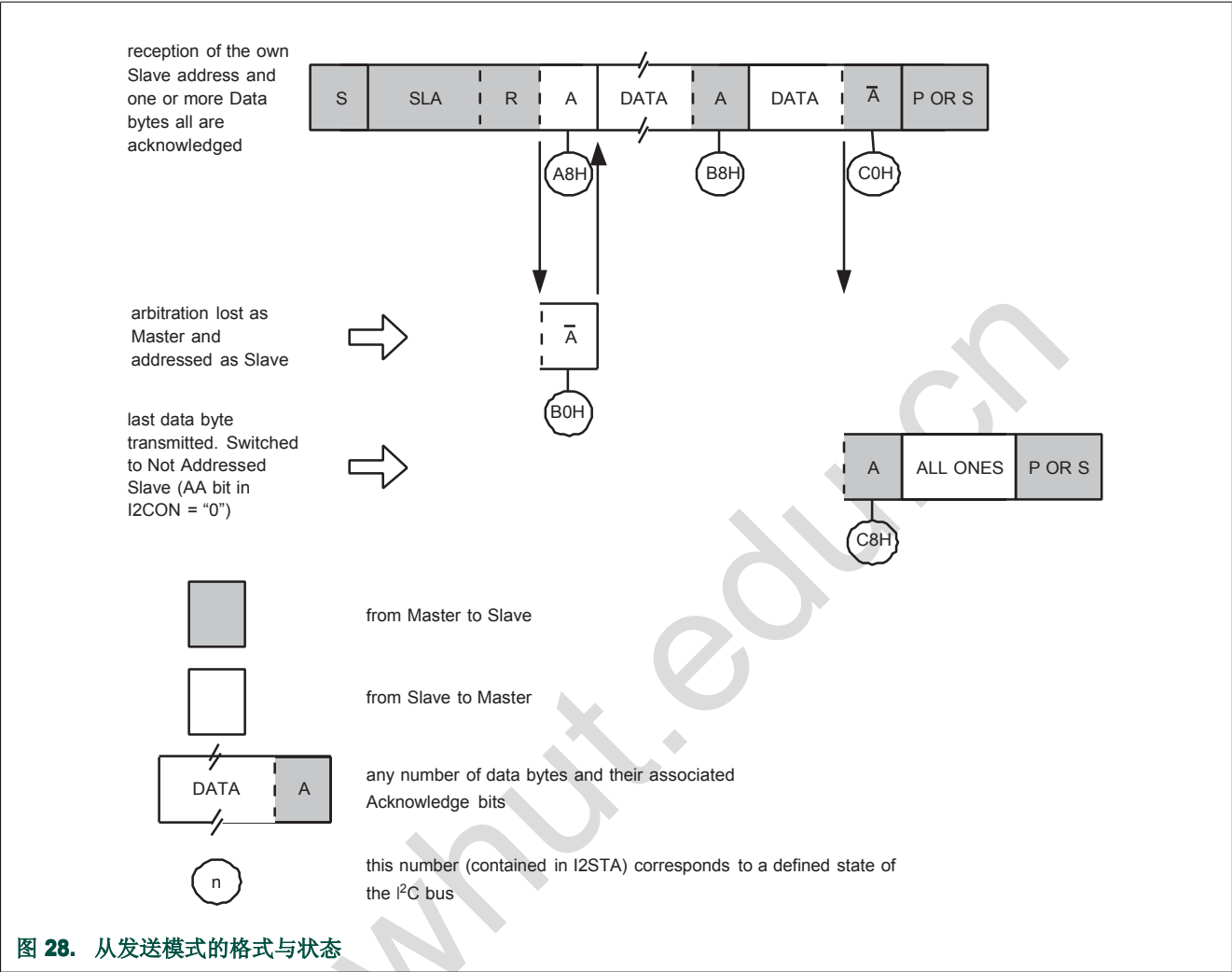
10.4 从发送模式

在从发送模式时，数据字节被传送到主接收器 (见 图10-28)。数据传输的初始化方法与从接收模式一样。当I2ADR和I2CON已被初始化之后，I²C模块将等待，直到它收到与其自身从地址相同的地址，且如果I²C模块要工作在从发送模式下，则随后的数据方向位必须是“1”(R)。在其从机地址和R位已被收到之后，串行中断标志(SI)被设置，而且可从I2STAT读取有效的状态代码。此状态代码是状态服务程序的向量，每一个状态代码对应合适的动作，具体动作详见表 10-162。如果I²C模块在主模式时仲裁丢失，也可进入从发送模式 (见状态码 0xB0)。

如果AA位在传输过程中复位，I²C模块将传输完最后一个字节，并进入的状态0xC0或0xC8。I²C模块切换到不可寻址匹配的从模式，如果它继续传输将忽略主接收器。因此，主接收器接收到的所有串行数据全是1。虽然AA复位，I²C模块将不响应自身从地址或广播地址。然而，I²C总线仍然被监测，且在任何时间可设置AA以恢复地址识别。这意味着，AA位可以被用来暂时将I²C块与I²C总线隔离。

表 160. 从发送模式

状态码 (I2CSTAT)	I2C总线及硬件状态	应用软件响应		I2C硬件的下一个动作			
		To/From I2DAT	To I2CON		AA		
			STA	STO			
0xA8	已接收到自身 SLA+R, 已返回ACK	加载数据字节 或	X	0	0	0	将传输最后一个数据字节; 将收到ACK。
		加载数据字节	X	0	0	1	将传输数据字节; 将收到ACK。
0xB0	作为主控设备, 在 SLA+R/W传输阶段, 仲裁丢失; 已接收到自 身 SLA+R, 已返回 ACK。	加载数据字节 或	X	0	0	0	将传输最后一个数据字节; 将收到ACK。
		加载数据字节	X	0	0	1	将传输数据字节; 将收到ACK。
		加载数据字节 或	X	0	0	0	将传输最后一个数据字节; 将收到ACK。
0xB8	I2DAT中的数据字节已 发送; 已接收到ACK。	加载数据字节	X	0	0	1	将传输数据字节; 将收到ACK。
0xC0	I2DAT中的数据字节已 发送; 已接收到NOT ACK。	无I2DAT动作 或	0	0	0	0	转换到不可寻址 SLV模式; 不能识别自身SLA 地址或广播地址。
		无I2DAT动作 或	0	0	0	1	转换到不可寻址 SLV模式; 将识别自身SLA 地 址, 如果I2ADR[0] = 1则识别广播地址。
		无I2DAT动作 或	1	0	0	0	转换到不可寻址 SLV模式; 不能识别自身 SLA 地址或广播地址。当总线空闲时, 将 发送一个START条件。
		无I2DAT动作	1	0	0	1	转换到不可寻址 SLV模式; 将识别自身 SLA 地址, 如果I2ADR[0] = 1则识别广播 地址。当总线空闲时, 将发送一个 START条件。
0xC8	I2DAT中的数据字节 已发送 (AA = 0); 已 接收到ACK。	无I2DAT动作 或	0	0	0	0	转换到不可寻址 SLV模式; 不能识别自身SLA 地址或广播地址。
		无I2DAT动作 或	0	0	0	1	转换到不可寻址 SLV模式; 将识别自身 SLA 地址, 如果I2ADR[0] = 1则识别广播 地址
		无I2DAT动作 或	1	0	0	0	转换到不可寻址 SLV模式; 不能识别自身 SLA 地址或广播地址。当总线空闲时, 将 发送一个START条件
		无I2DAT动作	1	0	0	01	转换到不可寻址 SLV模式; 将识别自身 SLA 地址, 如果I2ADR[0] = 1则识别广 播地址。当总线空闲时, 将发送一个 START条件



10.5 杂项状态

还有两个I2STAT代码, 不符合I²C硬件定义的状态(见 表 10–161)。这些将在下面讨论。

10.5.1 I2STAT = 0xF8

此状态代码表示没有相关信息可用, 因为串行中断标志位SI没有被设置。当I²C模块不参与串行传输时, 这种情况出现在其他状态之间。

10.5.2 I2STAT = 0x00

此状态代码表示在I²C串行传送期间发生了总线错误。START或STOP条件出现在帧格式中的非法位置, 就会产生一个总线错误。此状态代码表示在I²C串行传送期间发生了总线错误。START或STOP条件出现在帧格式中的非法位置, 就会产生一个总线错误。非常位置可以是在串行传输中的地址字节、数据字节或应答位中。当外部干扰信号干扰内部的I²C模块信号时, 也可能出现总线错误。当总线错误时, SI被设置。要恢复总线错误, 必须置STO标志位为1、且SI位必须被清除。这将导致的I2C块进入“不可寻址”的从模

式（一个定义的状态），并清除STO标志位（不影响I2CON其他位）。SDA和SCL线被释放（没有传输STOP条件）

表 161. 杂项状态

状态码 (I2CSTAT)	I2C总线及硬件状态	应用软件的响应 To/From I2DAT	To I2CON			AA	I2C硬件的下一个动作
			STA	STO			
0xF8	无相关的状态码可用； SI = 0.	无I2DAT动作	无I2CON动作				等待或进行当前传输。
0x00	由于非法的START或TOP条件，导致总线在MST或选定的从模式发生总线错误。在当干扰信号导致的I2C模块进入一个未定义状态时，也可能进入0x00状态。	无I2DAT动作	0	1	0	X	在MST或可寻址SLV模式下，仅内部硬件受影响。在所有的情况下，总线被释放，I2C模块进入不可寻址SLV模式，STO被复位。

10.6 一些特殊情况

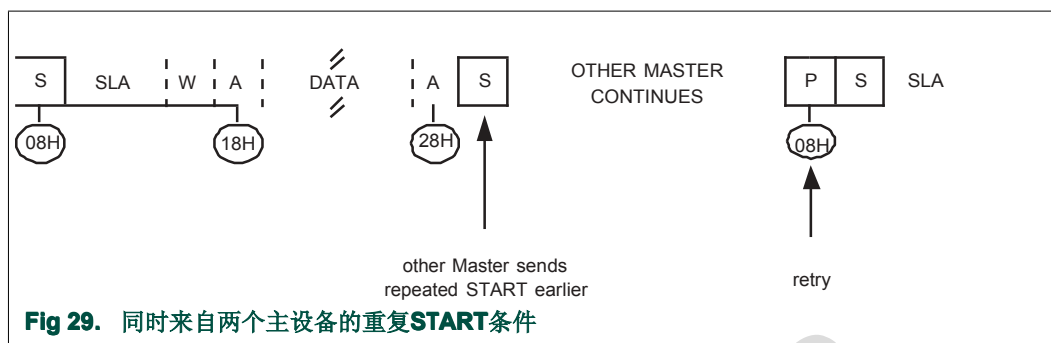
I2C硬件中有器件能处理在串行传输中可能的发生以下列特殊情况：

- 同时有来自两个主设备的重复START条件
- 仲裁丢失之后的数据传输
- 强制访问 I2C总线
- I2C总线被SCL或 SDA 线上的低电平阻塞
- 总线错误

10.6.1 同时有来自两个主设备的重复START条件

在主发送或主接收模式下，可能生成一个重复START条件。如果另一个主设备同时生成一个重复START件 (见 图 10–29)，则将会发生一个特殊的情况。发生这种情况之前，任何一方都没有丢失仲裁，因为它们传输的是相同的数据。

在I2C硬件本身产生重复START条件之前，如果在I2C总线上检测到一个重复START条件，它将释放总线，且不产生中断请求。如果另一个主设备通过产生STOP条件释放了总线，I2C模块将传输正常的START条件（状态0x08），并重新开始进行串行数据传输。



10.6.2 仲裁丢失后的数据传输

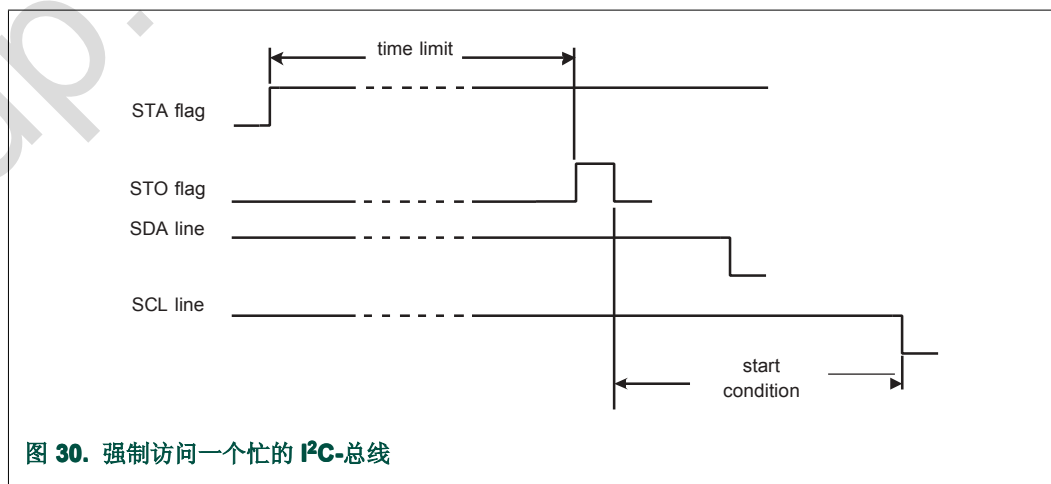
在主发送模式和主接收模式下，仲裁可能丢失（见图 10-23）。当I2STAT是以下状态仲裁可能丢失：0x38、0x68、0x78和 0xB0（见图 10-25 和 图 10-26）。

如果I2CON 中的STA标志被这些状态的服务程序设置，那么，如果总线再次空闲，在没有CPU的干预下，将传输一个START条件（状态0x08），并重新开始一个总串行传送。

10.6.3 强制访问 I²C总线

在一些应用中，可能存在一个不受控制的来源导致总线挂断。在这种情况下，问题可能是由干扰、总线上暂时的中断或在SDA和SCL之间临时性的短期的电流造成的。

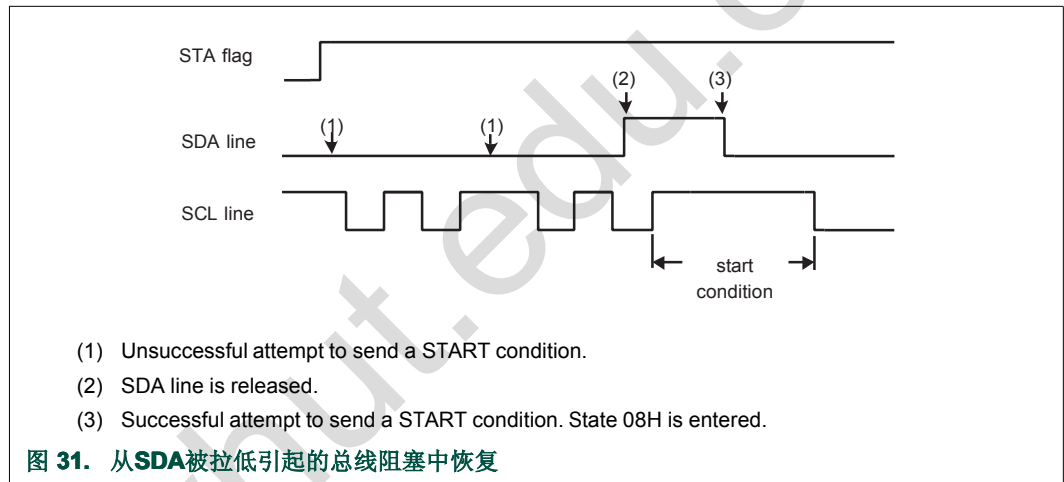
如果不受控制源产生了一个多余的START条件或屏蔽了STOP条件，那么I²C总线将一直保持忙碌状态。如果STA的标志被设置，且没有在合理的时间内访问总线，那么可能强制访问I²C总线。在STA状态标志仍被设置的情况下设置STO，就会出现强制访问，没有STOP条件被传送。此时，I²C硬件的行为就像收到STOP条件一样，将可以发送一个START条件。STO标志由硬件清除（见图 10-30）。



10.6.4 I²C总线被SCL或SDA线上的低电平阻塞

总线上的任何设备将SDA和SCL线上的电平拉低，就会发生一个I²C总线挂断。如果SCL线被总线上的设备阻塞（拉低），那么将不可能进行后续的串行传输，并且这个问题必须由拉低SCL线的设备解决。

通常情况下，SDA线在以下情况下可能会被阻塞：总线上的其他设备没有和当前的总线主控设备同步、任何错误的时钟或将噪声脉冲作为时钟。在这种情况下，问题可以通过在SCL线上传输一个额外时钟脉冲来解决(见 图 10-31)。I²C接口中没有一个专门的超时计时器来检测总线受阻，但这可以通过使用系统中另一个计时器来实现。当检测到SDA阻塞时，软件可以强制在SCL线上产生时钟（最多可达9个时钟周期），直到SDA被有问题的设备释放。在这一点上，从设备仍可能不同步，因此应产生一个START条件以保证所有的I²C外设是同步的。



10.6.5 总线错误

当START或STOP条件出现在帧格式中非法的位置上时，就会产生一个总线错误。串行传输中，非法位置有：地址字节、数据字节或应答位。

只有在I²C模块作为主设备或从设备并参与串行传输出现总线错误时，I²C硬件才会作出反应。当检测总线错误后，I²C模块立即切换到不可寻址从机模式，释放SDA和SCL线，设置中断标志，并用0x00装载状态寄存器。此状态码可用状态服务程序的向量，状态服务程序要么企图中止一次串行传输，要么简单地从错误状态中恢复（见表 10-163）。

10.7 I²C 状态服务程序

本节提供的操作例子，由不同I²C状态服务例程执行。这包括：

- 在复位后，初始化I²C块。
- I²C 中断服务。
- 支持所有4个I²C工作模式的26个状态服务程序。

10.8 初始化

在初始化的例子中，I²C模块可以为主机和从机模式。对于每个模式，都要有一个缓冲区用于发送和接收。初始化程序执行以下功能：

- 往I2ADR中加载模块自身的从地址和广播位 (GC)
- 允许 I²C 中断并设置中断优先级
- 同时设置I2CON中的I2EN和SI位，允许从设备模式；通过I2SCLH和I2SCLL寄存器定义串行时钟频率（对主模式。主设备程序必须在主程序中开始。

I²C 硬件开始在I²C总线上检测自身从地址和广播位。如果检测到广播地址或自身从地址被，将产生中断请求，并且在I2STAT中加载了一个合适的状态信息。

10.9 I²C 中断服务

当进入I²C中断，I2STAT包含一个状态代码，该状态代码确定26个状态服务中的哪个将被执行。

10.10 状态服务程序

每个状态例程是I²C中断例程的一部分，并处理26个状态之一。

10.11 将状态服务改为应用程序

状态服务的例子表明，响应26个I²C状态代码，必须一个执行典型动作。如果4个I²C操作模式中一个或更多模式没有被使用，相关的状态服务可被忽略，只要注意确认这些状态永远不会发生。

应用程序中，在I²C操作期间，可能需要执行某种超时程序，以处理总线无效或服务程序丢失。

11. 软件举例

11.1 初始化程序

初始化I²C 接口为主/从模式的例子。

1. 将自身从地址加载到I2ADR，如果需要，允许广播的识别。
2. 允许 I²C 中断。
3. 写 0x44 到I2CONSET以设置 I2EN 和 AA 位，允许从设备功能。仅是主设备功能，则写0x40 到 I2CONSET。

11.2 开始主设备传输功能

通过设置缓冲区、指针和数据计数，以开始主发送操作，然后启动一个START条件

1. 初始化主设备数据计数器。

2. 设置将要发送的从地址，再加上一个写（W）位。
3. 写 0x20 到 I2CONSET 以设置STA位。
4. 将要发出的数据写到主设备发送缓冲区中。
5. 初始化主数据计数器，以确定将被发送的消息的长度。
6. 退出。

11.3 开始主接收功能

通过设立缓冲区、指针和数据计数，以开始主接收操作，然后启动一个START条件。

1. 初始化主设备数据计数器。
2. 设置将要发送的从地址，再加上一个读（R）位。
3. 写 0x20 到 I2CONSET 以设置STA 位。
4. 设置主接收缓冲区。
5. 初始化主数据计数器，用于确定将被接收消息的长度。
6. 退出。

11.4 I²C 中断服务程序

确定 I²C 状态以及将要执行哪一个状态服务程序。

1. 从I2STA中读取I2C的状态。
2. 根据状态值，跳转到26种可能的状态服务程序之一。

11.5 无具体模式的状态

11.5.1 状态: 0x00

总线错误。进入不可寻址从模式并释放总线。

1. 写 0x14 到 I2CONSET 以设置 STO 和 AA 位。
2. 写 0x08 到 I2CONCLR 以清除 SI 标志。
3. 退出。

11.5.2 主设备状态

主接收模式和主发送模式下都会出现状态08和状态10。R / W位决定了下一个状态是主发送模式还是主接收模式。

11.5.3 状态: 0x08

已经发送一个START条件。将要发送从设备地址 + R/W 位，并将接收一个 ACK 位。

1. 写从地址和 R/W 位 到 I2DAT。
2. 写 0x04 到 I2CONSET 以设置AA 位。
3. 写 0x08 到 I2CONCLR以清除 SI 标志。
4. 设置主发送模式缓冲区。

5. 设置主接收模式缓冲区。
6. 初始化主数据计数器。
7. 退出。

11.5.4 状态: 0x10

已发送一个重复START条件。将要发送从设备地址 + R/W 位， 并接收一个ACK 位。

1. 写从地址和 R/W 位 到 I2DAT。
2. 写 0x04 到 I2CONSET 以设置AA 位。
3. 写 0x08 到 I2CONCLR以清除 SI 标志。
4. 设置主发送模式缓冲区。
5. 设置主接收模式缓冲区。
6. 初始化主数据计数器。
7. 退出。

11.6 主发送状态

11.6.1 状态: 0x18

前一次的状态是状态 8 或状态10，已完成从地址 + Write位传输， 已收到ACK。
将要发送第一个数据， 并接收ACK 位。

1. 从主发送模式缓冲区加载第一个数据到 I2DAT。
2. 写 0x04 到 I2CONSET 以设置AA 位。
3. 写 0x08 到 I2CONCLR以清除 SI 标志。
- 4.主发送缓冲区指针加1。
5. 退出。

11.6.2 状态: 0x20

已经完成从地址 + Write传输， 已接收到ACK。 将发送一个STOP条件。

1. 写 0x14 到 I2CONSET 以设置 STO 和 AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 退出。

11.6.3 状态: 0x28

数据传输完成，已接收到ACK。 如果传输的数据是最后一个数据字节，那么传输一个 STOP 条件， 否则传输下一个数据。

1. 主数据计数器减1，如果不是最后一个数据字节则跳到步骤5。
2. 写 0x14 到 I2CONSET 以设置 STO 和 AA 位。
3. 写 0x08 到 I2CONCLR以清除 SI 标志。
4. 退出。
5. 从主发送模式缓冲区加载下一个数据到 I2DAT。

6. 写 0x04 到 I2CONSET 以设置AA 位。
7. 写 0x08 到 I2CONCLR以清除 SI 标志。
8. 主发送缓冲区指针加1。
9. 退出。

11.6.4 状态: 0x30

数据传输完成，接收到NOT ACK。将传输一个 STOP 条件。

1. 写 0x14 到 I2CONSET 以设置 STO 和 AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 退出。

11.6.5 状态: 0x38

在传输从地址 + Write或数据时仲裁丢失。总线已经释放，并已进入不可寻址的从模式。当总线再次空闲时，将传输一个START 条件。

1. 写 0x24 到 I2CONSET 以设置 STA 和 AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 退出。

11.7 主接收状态

11.7.1 状态: 0x40

前一次状态是状态08或状态10。已经完成从地址+Read传输， 已返回ACK。将接收到数据并返回ACK。

1. 写 0x04 到 I2CONSET 以设置AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 退出。

11.7.2 状态: 0x48

已完成从地址+Read的传输， 已接收到NOT ACK。将传输一个STOP条件。

1. 写 0x14 到 I2CONSET 以设置 STO 和 AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 退出。

11.7.3 状态: 0x50

已接收到数据， 已返回ACK。将从I2DAT读出数据。将接收到额外的数据。如果这是最后一个数据字节则将返回NOT ACK， 否则将返回ACK。

1. 将I2DAT中的数据读出， 存到主接收模式缓冲区。
2. 主数据计数器减1， 如果不是最后一个数据字节跳到步骤5。
3. 写 0x0C 到 I2CONCLR 以清除 SI 标志和AA 位。

4. 退出
5. 写 0x04 到 I2CONSET 以设置AA 位。
6. 写 0x08 到 I2CONCLR以清除 SI 标志。
7. 主接收缓冲区的指针加1。
8. 退出。

11.7.4 状态: 0x58

已收到数据，已返回NOT ACK。将从I2DAT读出数据。将传输一个 STOP 条件。

1. 将I2DAT中的数据读出，存到主接收模式缓冲区。
2. 写 0x14 到 I2CONSET 以设置 STO 和 AA 位。
3. 写 0x08 到 I2CONCLR以清除 SI 标志。
4. 退出

11.8 从接收状态

11.8.1 状态: 0x60

已收到自身从地址 + Write位，已返回ACK。将接收数据并返回ACK。

1. 写 0x04 到 I2CONSET 以设置AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 设置从接收器模式数据缓冲区。
4. 初始化从设备数据计数器。
5. 退出。

11.8.2 状态: 0x68

作为总线主控器时，在自身从地址+R/W中阶段仲裁丢失。已收到自身从地址+Write。将接收到数据并返回ACK。当总线再次空闲时，设置STA位以重新启动主模式。

1. 写 0x24 到 I2CONSET以设置 STA 和 AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 设置从接收器模式数据缓冲区。
4. 初始化从设备数据计数器。
5. 退出。

11.8.3 状态: 0x70

已经接收到广播信息，ACK已返回。将接收数据并返回ACK。

1. 写 0x04 到 I2CONSET 以设置AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 设置从接收器模式数据缓冲区。

4. 初始化从设备数据计数器。
5. 退出

11.8.4 状态: 0x78

作为总线主控器时，在自身从地址+R/W传输阶段仲裁丢失。已收到广播地址，已返回ACK。将要接收数据并返回ACK。在总线再次空闲时，设置STA位以重新启动主模式。

1. 写 0x24 到 I2CONSET 以设置 STA 和 AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 设置从接收器模式数据缓冲区。
4. 初始化从设备数据计数器。
5. 退出。

11.8.5 状态: 0x80

前一次使用自身从地址寻址，数据已收到，已返回ACK。将要读额外的数据。

1. 将I2DAT中的数据读出，存到从接收模式缓冲区。
2. 从数据计数器减1，如果不是最后一个数据字节跳到步骤5。
3. 写 0x0C 到 I2CONCLR，以清除 SI 标志和AA 位。
4. 退出。
5. 写 0x04 到 I2CONSET 以设置AA 位。
6. 写 0x08 到 I2CONCLR以清除 SI 标志。
7. 从接收缓冲区指针加1。
8. 退出。

11.8.6 状态: 0x88

前一次使用自身从地址寻址，数据已接收到，已返回NOT ACK位。不保存接收到的数据。将进入不可寻址的从模式。

1. 写 0x04 到 I2CONSET 以设置AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 退出。

11.8.7 状态: 0x90

前一次寻址使用广播地址。数据已收到，已返回ACK位。将保存接收到的数据。只有第一个数据字节接收时返回ACK，其他数据字节接收将返回NOT ACK。

1. 将I2DAT中的数据读出，存到从接收模式缓冲区。
2. 写 0x0C 到 I2CONCLR以清除SI 标志和AA 位。
3. 退出。

11.8.8 状态: 0x98

前一次寻址使用广播地址。数据已收到，已返回NOT ACK位。不保存接收到的数据。将进入不可寻址的从模式。

1. 写 0x04 到 I2CONSET 以设置AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 退出。

11.8.9 状态: 0xA0

当仍作为一个可寻址从模式时，接收到一个STOP 条件或重复START条件。不保存接收到的数据。将进入不可寻址的从模式。

1. 写 0x04 到 I2CONSET 以设置AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 退出。

11.9 从发送状态

11.9.1 状态: 0xA8

已接收到自身从地址+读位，已返回 ACK。将要发出数据并将收到ACK位。

1. 从从发送模式缓冲区加载第一个数据到 I2DAT
2. 写 0x04 到 I2CONSET 以设置AA位。
3. 写 0x08 到 I2CONCLR以清除 SI标志。
4. 设置从发送模式数据缓冲区。
5. 从发送缓冲区指针加1。
6. 退出。

11.9.2 状态: 0xB0

作为总线主控器时，在自身从地址+R/W传输阶段仲裁丢失。已收到自身从地址+Read，已返回ACK。将要发送数据并返回ACK。在总线再次空闲时，设置STA位以重新启动主模式。

1. 从从发送模式缓冲区加载第一个数据到 I2DAT。
2. 写 0x24到I2CONSET以设置STA和AA位。
3. 写 0x08到I2CONCLR以清除SI标志。
4. 设置从发送器模式数据缓冲区。
5. 从发送缓冲区指针加1。
6. 退出。

11.9.3 状态: 0xB8

数据已经传输完成，已经返回ACK。将要发送数据，并接收ACK。

1. 从从发送模式缓冲区加载数据字节到I2DAT。

2. 写 0x04 到 I2CONSET 以设置AA位。
3. 写 0x08 到 I2CONCLR以清除 SI 标志。
4. 从发送缓冲区的指针加1。
5. 退出。

11.9.4 状态: 0xC0

数据已经传输完成，已经返回NOT ACK。进入不可寻址的从模式。

1. 写 0x04 到 I2CONSET 以设置AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 退出。

11.9.5 状态: 0xC8

最后一个数据字节已经传输完成，已返回ACK。进入不可寻址的从模式。

1. 写 0x04 到 I2CONSET 以设置AA 位。
2. 写 0x08 到 I2CONCLR以清除 SI 标志。
3. 退出。